THEORY OF COMPUTING, Volume 11 (5), 2015, pp. 149–182
www.theoryofcomputing.org

List-Decoding Multiplicity Codes

Swastik Kopparty*

Received February 4, 2013; Revised April 1, 2015; Published May 29, 2015

Abstract: We study the list-decodability of multiplicity codes. These codes, which are based on evaluations of high-degree polynomials and their derivatives, have rate approaching 1 while simultaneously allowing for sublinear-time error correction. In this paper, we show that multiplicity codes also admit powerful list-decoding and local list-decoding algorithms that work even in the presence of a large error fraction. In other words, we give algorithms for recovering a polynomial given several evaluations of it and its derivatives, where possibly many of the given evaluations are incorrect.

Our first main result shows that univariate multiplicity codes over fields of prime order can be list-decoded up to the so-called "list-decoding capacity." Specifically, we show that univariate multiplicity codes of rate R over fields of prime order can be list-decoded from a $(1-R-\varepsilon)$ fraction of errors in polynomial time (for constant R,ε). This resembles the behavior of the "Folded Reed-Solomon Codes" of Guruswami and Rudra (Trans. Info. Theory 2008). The list-decoding algorithm is based on constructing a differential equation of which the desired codeword is a solution; this differential equation is then solved using a power-series approach (a variation of Hensel lifting) along with other algebraic ideas.

Our second main result is a list-decoding algorithm for decoding multivariate multiplicity codes up to their Johnson radius. The key ingredient of this algorithm is the construction of a special family of "algebraically-repelling" curves passing through the points of $\mathbb{F}_q^m$; no moderate-degree multivariate polynomial over $\mathbb{F}_q^m$ can simultaneously vanish on all these

A version of this paper was posted online as an Electronic Colloq. on Computational Complexity Tech. Report [20].

*Supported in part by a Sloan Fellowship and NSF grant CCF-1253886.

ACM Classification: E.4, G.1.7

AMS Classification: 11T71, 94B35, 68W30

Key words and phrases: error-correcting codes, list-decoding, polynomials, differential equations

DOI: 10.4086/toc.2015.v011a005

curves. These curves enable us to reduce the decoding of multivariate multiplicity codes over $\mathbb{F}_q^m$ to several instances of decoding univariate multiplicity codes over the big field $\mathbb{F}_{q^m}$, for which such list-decoding algorithms are known.

As a corollary, we show how multivariate multiplicity codes of length n and rate nearly 1 can be locally list-decoded up to their Johnson radius in $O(n^{\varepsilon})$ time.

1 Introduction

Reed-Solomon codes (which are codes based on univariate polynomials) and Reed-Muller codes (which are codes based on multivariate polynomials) are classical families of error-correcting codes which have found wide applicability within theoretical computer science. One of the crucial reasons for these codes being very useful, apart from the fact that polynomials appear naturally in many basic settings, is that they admit very good decoding algorithms.

In recent work [22], a new family of polynomial codes called multiplicity codes was introduced. These codes extend the classical polynomial codes; they are based on evaluating polynomials *and their derivatives*. The codewords of the m-variate order-s multiplicity code of degree-d polynomials over $\mathbb{F}_q$ are obtained by taking an m-variate polynomial over $\mathbb{F}_q$ of degree at most d, finding all its derivatives up to order s, and evaluating all of them on all the points of $\mathbb{F}_q^m$. A typical setting of interest is where $\mathbb{F}_q$ is a large finite field, m, s are constants, and $d = (1-\delta)sq$ for some $\delta \in (0,1)$. (So δ becomes the minimum distance of this code.) The augmentation of the derivatives allows one to consider polynomials over $\mathbb{F}_q$ of degree larger than q, and this leads to much better tradeoffs in the rate and minimum distance of these codes, while retaining the good local decodability of Reed-Muller codes.

In this work, we study the list-decodability of multiplicity codes. Specifically, we study the problem of decoding multiplicity codes from an α fraction of errors: given a "received" vector, the problem is to find the list of all codewords of the multiplicity code which are within distance α from this vector. We show two main results, showing that multiplicity codes of a given rate can be list-decoded from a much larger fraction of errors than what is known for the corresponding classical code.

- Our first result shows that univariate multiplicity codes achieve "list-decoding capacity." Specifically, for every R, ε, there is a family of univariate multiplicity codes of rate R which can be list-decoded from a $(1-R-\varepsilon)$ fraction of errors in polynomial time. This thus provides a new (and perhaps more natural) example of an explicit capacity-achieving list-decodable code, the only other example being the original "Folded Reed-Solomon codes" of Guruswami and Rudra [10].
- Our second result shows how to list-decode multivariate multiplicity codes up to the Johnson bound. Specifically, we give a polynomial time algorithm for list-decoding multivariate multiplicity codes of minimum distance δ from a $(1-\sqrt{1-\delta})$ fraction of errors. In particular, this gives the first algorithm for unique-decoding multivariate multiplicity codes up to half the minimum distance of the code.

As a corollary of our second result, we show that m-variate multiplicity codes of distance δ (which have rate $\approx (1-\delta)^m$) can be *locally* list-decoded from a $(1-\sqrt{1-\delta}-\varepsilon)$ fraction of errors in time $O(n^{O(1/m)})$. This also gives the first algorithm for local decoding of multivariate multiplicity codes up to half the minimum distance. (The local decoding algorithm of [22] can at best decode from $1/4$ the minimum distance.) This gives the best known tradeoff between rate, error tolerance, and query complexity of local decoding.

One goal of this line of research is to develop good algorithms to support error correction for multiplicity codes. Multiplicity codes are currently the only known codes which achieve rate arbitrarily close to 1 while supporting $o(n)$-time local decodability. Thus they could potentially be useful for real-life error correction. In addition, these codes are very closely related to the omnipresent (and omnipotent?) polynomial codes, and they seem ripe for use in complexity theory and pseudorandomness applications.

The other motivation is that the actual algorithmic questions underlying these decoding questions are very natural and basic; they amount to interpolating a polynomial given evaluations of it and its derivatives at various points, even though some of the given evaluations may be wrong.

We now describe our results and methods in detail.

1.1 List-decoding of univariate multiplicity codes

Our first result deals with univariate multiplicity codes. It is well known that for every $R \in (0,1)$, the largest fraction of errors from which an error-correcting code of rate R can be list-decoded, with polynomial size lists, is less than $(1-R)$. We show that univariate multiplicity codes of rate R achieve "list-decoding capacity"; they can be list-decoded from a $(1-R-\varepsilon)$ fraction of errors in polynomial time with polynomial size lists. The first (and only known) example of such codes was given by Guruswami and Rudra [10].

Theorem A (Informal)**.** For every $R \in (0,1)$ and every $\varepsilon > 0$, there is an $s > 0$ such that for every sufficiently large prime q, the order-s univariate multiplicity code over $\mathbb{F}_q$ of length n and rate R can be list-decoded from a $(1-R-\varepsilon)$ fraction of errors in time $\mathrm{poly}(n)$.

See Theorem 3.1 for a formal statement of this result.

At the core of this theorem is an algorithm for the following natural problem: given n tuples $(\alpha_i, \beta_i^{(0)}, \beta_i^{(1)}, \ldots, \beta_i^{(s-1)})$ in $\mathbb{F}_q^{s+1}$, find all polynomials $P(T)$ of degree at most d such that for at least A values of $i \in [n]$, we have $P^{(j)}(\alpha_i) = \beta_i^{(j)}$ for each $j < s$ (here $P^{(j)}$ denotes the j'th derivative of P).

Following earlier algorithms for list-decoding algebraic codes [25, 12, 23, 10], our algorithm is based on first processing the tuples to find an equation which we know every such $P(T)$ satisfies, and then finding all solutions of that equation.

For multiplicity codes the kind of equation that turns out to be appropriate is a differential equation. Specifically we first use the tuples to find a differential equation of the form

$$Q(T, P(T), P^{(1)}(T), \ldots, P^{(r)}(T)) = 0$$

which every such $P(T)$ satisfies, where Q is a multivariate low-degree polynomial. This equation is found by imposing constraints on Q that force it to vanish with a suitable "weighted-multiplicity" along certain curves that come from the tuples.

Once we have done this, we need to solve the differential equation. Fortunately, it turns out that there are classical techniques that can give us a grip on such equations. Next we give an overview of these classical techniques and how these can be used to find all low-degree polynomials $P(T)$ which are solutions of the differential equation.

1.1.1 Interlude: The Cauchy-Kowalevski differential equation

In characteristic 0, the differential equation

$$Q(T, P(T), P'(T)) = 0$$

and its higher-order analogues, are well studied in analysis and mathematical physics. If $P(T)$ is assumed to be analytic, then this is called the Cauchy-Kowalevski differential equation. One example of a situation where such equations arise is the following: Let T be "time" and let $P(T)$ be the "position of some particle at time T." Then $P'(T)$ is the velocity at time T. The equation above expresses some invariant maintained in the evolution of the particles position and velocity over time. The familiar "$(1/2)mv^2 + mgh = C$" is an example of such a differential equation.

The classical analysis approach to such equations (see, e. g., [5]) is based on power series expansions. To begin, suppose one is given some initial conditions $T = t_0$, $P(t_0) = \alpha_0$, $P'(t_0) = \alpha_1$. If these initial conditions are "nonsingular" (which roughly means that the initial conditions uniquely specify a particular solution of the differential equation), one can simply recover $P(T)$ as a power series in $T - t_0$ by iteratively considering the equation mod $(T - t_0)$, mod $(T - t_0)^2$, mod $(T - t_0)^3$, etc.; at each iteration we get one new coefficient of the power series representation of $P(T)$. This method works essentially verbatim even when we work over finite fields (over $\mathbb{C}$, the nontrivial part is to show that this power series is convergent, but this does not bother us when we work with formal power series).

We thus know how to find the solution $P(T)$ if we are given nonsingular initial conditions consistent with $P(T)$. This motivates the following algorithm: try various choices of the initial conditions, and whenever the initial conditions are nonsingular, recover the power series expansion of the solution with the given initial conditions. For any given $P(T)$, if we chance upon initial conditions that are nonsingular and consistent with $P(T)$, this algorithm will recover $P(T)$.

Quite unfortunately, there can be "very singular" situations; $P(T)$ could be such that no matter which initial conditions we consider, those initial conditions are singular. At this point an algebraic phenomenon specific to our setting (and nonexistent in the analytic setting) comes to the rescue. It turns out that in the case where $P(T)$ is a low degree polynomial, which is the case we are interested in, the very singular situation happens only when $P(T)$ also satisfies a certain related differential equation $Q^*(T, P(T), P'(T)) = 0$. Furthermore, Q^* is of lower degree; and thus we have reduced our problem to that of solving a simpler differential equation. Iterating this, we arrive at the complete list of all the low degree $P(T)$ that satisfy the original differential equation.

1.2 List-decoding of multivariate multiplicity codes

Our second result deals with general multiplicity codes. Recall that the Johnson bound states that for every code of distance δ, the list-size for list-decoding from a $(1 - \sqrt{1-\delta})$ fraction of errors is at most a polynomial in the length of the code. We show that for multiplicity codes, this can be made algorithmic; we can find that list in polynomial time.

Theorem B (Informal)**.** For $m, s = O(1)$, every m-variate order-s multiplicity code of length n and distance δ can be list-decoded from a $(1 - \sqrt{1-\delta})$ fraction of errors in time poly(n).

See Theorem 3.2 for a formal statement of this result.

When $s = 1$ (the classical polynomial code case), this result was previously shown by Guruswami-Sudan [12] and Pellikaan-Wu [24]. When $m = 1$ (the univariate case), this result was previously shown by Guruswami-Sudan [12] and Guruswami-Sahai-Sudan [11] (this is based on the fact that univariate multiplicity codes are a special case of ideal-based error-correcting codes [26]). Also note that when $m = 1$ and q is prime, Theorem A is stronger than Theorem B.

Theorem B is also the first algorithm for decoding multivariate multiplicity codes up to half the minimum distance.

All the known algorithms for decoding classical multivariate polynomial codes up to half their minimum distance are quite indirect and specialized.[1] We briefly recall one of these algorithms, which proceeds via a reduction to decoding univariate polynomial codes. Here one is given a received word $r : \mathbb{F}_q^m \to \mathbb{F}_q$, and one wants to find a multivariate polynomial $Q(X_1, \ldots, X_m)$ of degree at most d such that $Q(\mathbf{a}) = r(\mathbf{a})$ for many $\mathbf{a} \in \mathbb{F}_q^m$. One first finds a special kind of low-degree curve γ which maps the finite field $\mathbb{F}_{q^m}$ bijectively to the vector space $\mathbb{F}_q^m$. Using γ, we get a function $r \circ \gamma : \mathbb{F}_{q^m} \to \mathbb{F}_q$. By the bijectivity of γ, every $Q(X_1, \ldots, X_m)$ which has high agreement with r will also have the univariate polynomial $Q \circ \gamma(T)$ having high agreement with $r \circ \gamma$. Since $Q \circ \gamma(T)$ is a low-degree univariate polynomial (by the low-degreeness of Q and γ), we have reduced to the problem of finding all univariate polynomial that has high agreement with $r \circ \gamma$. This is precisely the problem of list-decoding univariate polynomial codes; tracing the parameters through, an algorithm for list-decoding univariate polynomial codes up to their Johnson radius gives an algorithm for decoding multivariate polynomial codes up to their Johnson radius (and the same statement holds for decoding up to half the minimum distance).

While dealing with multiplicity codes, the possibility that the multivariate polynomials we are dealing with have degree $> q$ starts causing difficulties. It turns out that there can be exponentially many polynomials $Q(X_1, \ldots, X_m)$ of degree $2q$ (say) such that the compositions $Q \circ \gamma(T)$ are all the same univariate polynomial. Thus composing with a curve γ no longer preserves the original decoding problem unambiguously.

To bypass this, we use a special family of "algebraically repelling" curves from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q^m$. More concretely, we will construct a collection of curves $\gamma_i : \mathbb{F}_{q^m} \to \mathbb{F}_q^m$ which have the property that for any low-degree polynomial $Q(X_1, \ldots, X_m)$ of degree less than sq, the collection of univariate polynomials $Q \circ \gamma_i$ uniquely determines Q. Equivalently, if Q formally vanishes on all the γ_i, then Q itself must be identically 0.

Given these curves, the decoding algorithm for multivariate multiplicity codes does the following. Using the received word

$$r : \mathbb{F}_q^m \to \mathbb{F}_q^{\binom{m+s-1}{m}}$$

for the multivariate multiplicity code, one comes up with a collection or received words

$$r_i : \mathbb{F}_{q^m} \to (\mathbb{F}_{q^m})^s$$

[1] To illustrate this, we note that it is unknown how to efficiently decode multivariate polynomial codes from half the minimum distance when the set of evaluation points is $S \times S$, where S is an arbitrary subset of $\mathbb{F}_q$; it is only known for certain special sets S such as $S = \mathbb{F}_q$. This is quite odd, since the principle on which the minimum distance of these codes is proved has a proof which does not depend on the specific choice of S.

(in spirit, r_i is the composition of r with γ_i) for a univariate multiplicity code. Then via a univariate multiplicity code decoder, we find low-degree univariate polynomials P_i which have high agreement with the r_i. Then using the above mentioned property of the γ_i, there is a unique $Q(X_1,\dots,X_m)$ such that $Q \circ \gamma_i = P_i$ for each i. Q is then our decoded multivariate polynomial.

1.3 Local decoding and local list-decoding of multiplicity codes

The above list-decoding algorithm for multivariate multiplicity codes (in particular the bivariate case) enables improved local decoding and local list-decoding algorithms for multiplicity codes.

Theorem C (Informal)**.** For every $\varepsilon > 0$ and integers $m, s = O(1)$, every m-variate order-s multiplicity code of length n and distance δ can be:

- locally decoded from a $(\delta/2 - \varepsilon)$ fraction of errors, and
- locally list-decoded from a $(1 - \sqrt{1-\delta} - \varepsilon)$ fraction of errors,

in $n^{O(1/m)}$ time, using $n^{O(1/m)}$ queries.

See Theorem 3.3 for a formal statement of this result.

The local decoding algorithm follows the original [22] approach, using planes instead of lines. The reason planes allows for better local decoding than lines is the following simple fact: for a fixed point $\mathbf{a} \in \mathbb{F}_q^m$ and a fixed set $S \subseteq \mathbb{F}_q^m$ (which will correspond to the erroneous coordinates), if we pick a uniformly random plane P passing through $\mathbf{a}$, with high probability P samples S well:

$$\frac{|P \cap S|}{|P|} \approx \frac{|S|}{|\mathbb{F}_q^m|}.$$

This fact allows us decode the corrected value of a corrupted multiplicity codeword at a point $\mathbf{a} \in \mathbb{F}_q^m$ by decoding several bivariate multiplicity codewords; this is where the algorithm of Theorem B gets used.

The local list-decoding algorithm for multiplicity codes follows the basic outline of the Arora-Sudan [1] and Sudan-Trevisan-Vadhan [27] local list-decoders for multivariate polynomial codes (specifically the analysis from [2], which uses decoding on planes).

1.4 Interpolating sets for multiplicity codes

Finally, we also give a construction of explicit interpolating sets for multiplicity codes.

Multiplicity codes are most naturally "locally-correctable." To make them locally-decodable in the strictest sense we need to specify an encoding map E, as well as a local decoding algorithm which when given access to a corrupted version of $E(m)$, finds any given symbol of the original message m. The existence of such an encoding map follows easily from the linearity of the code. Such a map can be efficiently computed with the help of advice which is of length polynomial in the length of the code. Thus the sublinear time local-correction algorithm of [22] implies that multiplicity codes have non-uniform local-decoding algorithms which run in sublinear time.

To make the local decoding fully uniform, we need to be able to explicitly describe the encoding map. This can be done by giving an interpolating set: a set of coordinates of the multiplicity code such that if we specify symbols arbitrarily on those coordinates, there is a unique codeword of the multiplicity code that extends this. We give an explicit description of an interpolating set for multiplicity codes.

Theorem D (Informal)**.** There are explicit interpolating sets for multiplicity codes. Thus, multiplicity codes can be locally decoded in uniform sublinear time.

See Section 6 for a description of the construction and an outline of the analysis.

The interpolating sets we find are simple combinations of standard interpolating sets for multivariate polynomial codes. They are based on representing high-degree multivariate polynomials as appropriate linear combinations of multivariate polynomials with individual degrees bounded by $q-1$.

1.5 Related and subsequent work

Theorem A was independently discovered by Guruswami and Wang [13]. The method of proof and the algorithm there are simpler than ours, but our approach yields a slightly better list-decoding radius for each fixed multiplicity code. An exact comparison appears in Section 3. (This is similar to the relationship that Vadhan's simpler version, see Chapter 5 of [28], of the list-decoding of Folded Reed-Solomon codes has to the original algorithm of Guruswami and Rudra [10].)

Since this paper was posted online [20], there have been a number of advances in the theory of list decoding. Many new families of codes meeting list-decoding capacity have been discovered [13, 4, 16, 14, 15, 9], and today we know constructions of such codes which also achieve near-optimal alphabet size, list size and list-decoding time. See [9] for a detailed description of the state of the art in this area.

There have also been new constructions of locally correctable codes of high rate [8, 6, 17], qualitatively matching the local correctability of multiplicity codes. Local list-decoding algorithms for the high-rate codes of [8] were given by [7]. Some further developments in the theory of multiplicity codes are discussed in the survey [21].

Organization of this paper. In the next section we introduce some notation related to codes, polynomials, derivatives, multiplicities and multiplicity codes. In Section 3 we state our main results formally. In Section 4 we prove Theorem A. In Section 5 we prove Theorem B. Theorem C and Theorem D are proved in Sections 6 and 7 respectively. We conclude with some open problems.

2 Preliminaries

2.1 Codes

Let Σ be a finite set and let n be an integer. We will endow Σ^n with a metric called the (normalized) Hamming distance Δ, defined by:

$$\Delta(x,y) = \Pr_{i\in[n]}[x_i \neq y_i].$$

A *code of length n over the alphabet* Σ is simply a subset $\mathcal{C}$ of Σ^n. The rate of the code is defined to be:

$$R = \frac{\log_{|\Sigma|}|\mathcal{C}|}{n}.$$

The minimum distance of the code $\mathcal{C}$ is defined to be the smallest value of $\Delta(c,c')$ for distinct elements c,c' of $\mathcal{C}$.

List decoding. In the problem of *list-decoding* the code $\mathcal{C}$ from an η fraction of errors, we are given as input $r \in \Sigma^n$, and we wish to compute the set

$$\mathcal{L} = \{c \in \mathcal{C} \mid \Delta(r,c) < \eta\}\,.$$

The maximum possible value of $|\mathcal{L}|$ as r varies over all elements of Σ^n is called the *list-size* for list-decoding $\mathcal{C}$ from an η fraction of errors.

2.2 Polynomials and derivatives

We use $[m]$ to denote the set $\{1,\dots,m\}$. For a vector $\mathbf{i} = \langle i_1,\dots,i_m\rangle$ of non-negative integers, its *weight*, denoted $\mathrm{wt}(\mathbf{i})$, equals $\sum_{j=1}^m i_j$.

For a field $\mathbb{F}$, let $\mathbb{F}[X_1,\dots,X_m] = \mathbb{F}[\mathbf{X}]$ be the ring of polynomials in the variables $X_1,\dots,X_m$ with coefficients in $\mathbb{F}$.

For a vector of non-negative integers $\mathbf{i} = \langle i_1,\dots,i_m\rangle$, let $\mathbf{X}^{\mathbf{i}}$ denote the monomial $\prod_{j=1}^m X_j^{i_j} \in \mathbb{F}[\mathbf{X}]$. The $(c_1,\dots,c_m)$-weighted degree of this monomial is defined to be $\sum_{j=1}^m c_j i_j$. The $(c_1,\dots,c_m)$-weighted degree of a polynomial is defined to be the maximum $(c_1,\dots,c_m)$-degree of any of its nonzero monomials. A special case of interest is the *total degree* of a monomial/polynomial, which is defined to be the $(1,1,\dots,1)$-weighted degree of that monomial/polynomial. We use the term degree to denote the total degree when there is no confusion.

We now define derivatives and the multiplicity of vanishing at a point, and then state a basic bound on the total number of zeroes (counting multiplicity) that a polynomial can have on a product set S^m. An elementary proof of this lemma can be found in [3].

Definition 2.1 ((Hasse) Derivative)**.** For $P(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ and non-negative vector $\mathbf{i}$, the $\mathbf{i}$th *(Hasse) derivative* of P, denoted $P^{(\mathbf{i})}(\mathbf{X})$, is the coefficient of $\mathbf{Z}^{\mathbf{i}}$ in the polynomial $\tilde{P}(\mathbf{X},\mathbf{Z}) \overset{\mathrm{def}}{=} P(\mathbf{X}+\mathbf{Z}) \in \mathbb{F}[\mathbf{X},\mathbf{Z}]$.

Thus,

$$P(\mathbf{X}+\mathbf{Z}) = \sum_{\mathbf{i}} P^{(\mathbf{i})}(\mathbf{X})\mathbf{Z}^{\mathbf{i}}. \tag{2.1}$$

Definition 2.2 (Multiplicity)**.** For $P(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ and $\mathbf{a} \in \mathbb{F}^n$, the *multiplicity* of P at $\mathbf{a} \in \mathbb{F}^n$, denoted $\mathsf{mult}(P,\mathbf{a})$, is the largest integer M such that for every non-negative vector $\mathbf{i}$ with $\mathrm{wt}(\mathbf{i}) < M$, we have $P^{(\mathbf{i})}(\mathbf{a}) = 0$ (if M may be taken arbitrarily large, we set $\mathsf{mult}(P,\mathbf{a}) = \infty$).

Lemma 2.3. *Let $P \in \mathbb{F}[\mathbf{X}]$ be a nonzero polynomial of total degree at most d. Then for any finite $S \subseteq \mathbb{F}$,*

$$\sum_{\mathbf{a}\in S^n} \mathsf{mult}(P,\mathbf{a}) \le d \cdot |S|^{n-1}\,.$$

In particular, for any integer $s > 0$,

$$\Pr_{\mathbf{a}\in S^n}[\mathsf{mult}(P,\mathbf{a}) \ge s] \le \frac{d}{s|S|}\,.$$

2.3 Multiplicity codes

Now we recall the definition of multiplicity codes and the formulas for their rate and minimum distance.

Definition 2.4 (Multiplicity code [22])**.** Let s, d, m be nonnegative integers and let q be a prime power. Let

$$\Sigma = \mathbb{F}_q^{\binom{m+s-1}{m}} = \mathbb{F}_q^{\{\mathbf{i}:\mathrm{wt}(\mathbf{i})<s\}} .$$

For $P(X_1,\dots,X_m) \in \mathbb{F}_q[X_1,\dots,X_m]$, we define *the order-s evaluation of P at* $\mathbf{a}$, denoted $P^{(<s)}(\mathbf{a})$, to be the vector $\langle P^{(\mathbf{i})}(\mathbf{a})\rangle_{\mathrm{wt}(\mathbf{i})<s} \in \Sigma$.

We define *the multiplicity code of order-s evaluations of degree-d polynomials in m variables over* $\mathbb{F}_q$ as follows. The code is over the alphabet Σ, and has length q^m (where the coordinates are indexed by elements of $\mathbb{F}_q^m$). For each polynomial $P(\mathbf{X}) \in \mathbb{F}_q[X_1,\dots,X_m]$ with $\deg(P) \le d$, there is a codeword in $\mathcal{C}$ given by:

$$\mathrm{Enc}_{s,d,m,q}(P) = \langle P^{(<s)}(\mathbf{a})\rangle_{\mathbf{a}\in\mathbb{F}_q^m} \in (\Sigma)^{q^m} .$$

Lemma 2.5 (Rate and distance of multiplicity codes [22])**.** *Let $\mathcal{C}$ be the multiplicity code of order-s evaluations of degree-d polynomials in m variables over $\mathbb{F}_q$. Then $\mathcal{C}$ has minimum distance*

$$\delta = 1 - \frac{d}{sq} \quad \textit{and rate} \quad \frac{\binom{d+m}{m}}{\binom{s+m-1}{m} q^m},$$

which is at least

$$\left(\frac{s}{m+s}\right)^m \cdot \left(\frac{d}{sq}\right)^m \ge \left(1 - \frac{m^2}{s}\right)(1-\delta)^m .$$

Note that in the $m = 1$ case, the rate of the code is $(d+1)/(sq) \ge 1-\delta$, and thus univariate multiplicity codes lie on the Singleton bound.

3 Main theorems

We first consider the problem of list-decoding univariate multiplicity codes.

Theorem 3.1 (List-decodability of Univariate Multiplicity Codes)**.** *Let r, s be integers with $0 \le r < s = O(1)$. Let $\delta \in (0,1)$ be a constant, and let $R = 1-\delta$.*

For every prime q, let $d = (1-\delta)sq$ and let $\mathcal{C}_q$ be the multiplicity code of order-s evaluations of degree-d polynomials in 1 variable over $\mathbb{F}_q$. Thus $\mathcal{C}_q$ has rate R and minimum distance δ.

There is an algorithm that can list-decode $\mathcal{C}_q$ from a

$$1 - \left(\frac{s}{s-r}\cdot R\right)^{\frac{r+1}{r+2}} - \max\left(\frac{16s^2}{d}, \frac{8s^2}{q}\right)$$

fraction of errors in time $q^{O(rs)}$ with lists of size $q^{O(rs)}$.

We prove this theorem in Section 4. Taking

$$r = \left\lceil \frac{\log(1+\frac{\varepsilon}{3})}{\log(1/R)} \right\rceil \quad \text{and} \quad s = r \cdot \left(1 + \frac{3}{\varepsilon}\right),$$

we have for sufficiently large q:

$$\begin{aligned} \left(\frac{s}{s-r} \cdot R\right)^{\frac{r+1}{r+2}} + \max\left(\frac{16s^2}{d}, \frac{8s^2}{q}\right) &\leq (1+\varepsilon/3) \cdot R^{1-1/r} + \varepsilon/3 \\ &\leq (1+\varepsilon/3) \cdot R \cdot (1+\varepsilon/3) + \varepsilon/3 \\ &\leq (1+2\varepsilon/3) \cdot R + \varepsilon/3 \\ &\leq R + \varepsilon . \end{aligned}$$

This gives the desired $1-R-\varepsilon$ list-decoding radius, and so we get Theorem A from the introduction.

The algorithm of Guruswami and Wang [13] can decode multiplicity codes from a

$$\frac{r+1}{r+2} \cdot \left(1 - \frac{s}{s-r} \cdot R\right)$$

fraction of errors. This decoding radius is smaller than the decoding radius of our algorithm (upto the vanishing term $\max(16s^2/d, 8s^2/q)$) by the AM-GM inequality.

Next we consider the problem of list-decoding multivariate multiplicity codes.

Theorem 3.2. *Let $s, m = O(1)$. Let $\delta \in (0,1)$ be a constant.*

For every prime power q, let $d = (1-\delta)sq$ and let $\mathcal{C}_q$ be the multiplicity code of order-s evaluations of degree-d polynomials in m variables over $\mathbb{F}_q$. Thus $\mathcal{C}_q$ has minimum distance δ.

There is an algorithm that can list-decode $\mathcal{C}_q$ from a

$$1 - \sqrt{1-\delta} - \varepsilon$$

fraction of errors in time $q^{O(1)}$, with lists of size $\mathrm{poly}(1/\varepsilon)$. *If $\varepsilon = 0$, the the list size is $q^{O(1)}$.*

The proof of this theorem appears in Section 5.

As a corollary of this, we show the local list-decodability of multivariate multiplicity codes.

Theorem 3.3. *Let $s, m = O(1)$. Let $\delta \in (0,1)$ be a constant.*

For every prime power q, let $d = (1-\delta)sq$ and let $\mathcal{C}_q$ be the multiplicity code of order-s evaluations of degree-d polynomials in m variables over $\mathbb{F}_q$. Thus $\mathcal{C}_q$ has minimum distance δ.

There is an algorithm that can locally list-decode $\mathcal{C}_q$ from a

$$1 - \sqrt{1-\delta} - \varepsilon$$

fraction of errors in time $O(q^6)$, using $O(q^2)$ queries, with lists of size $\mathrm{poly}(1/\varepsilon)$.

The proof of this theorem is sketched in Section 7.

4 List decoding univariate multiplicity codes

We first describe a slightly more general problem than that problem of list-decoding univariate multiplicity codes, and we give our algorithmic results for this general problem. Later we will specialize our results to univariate multiplicity codes.

Let
$$S = \left\{ \left(\alpha_i, \beta_i^{(0)}, \beta_i^{(1)}, \ldots, \beta_i^{(s-1)}\right) \;\middle|\; i \in [n] \right\}$$
be a collection of n distinct points in $\mathbb{F}_q^{s+1}$. We will use $\beta_i^{(<s)}$ to denote $(\beta_i^{(0)}, \beta_i^{(1)}, \ldots, \beta_i^{(s-1)}) \in \mathbb{F}_q^s$.

We are interested in polynomials $P(T) \in \mathbb{F}_q[T]$ of degree-d which have "high agreement" with S. Define:
$$\mathrm{agree}(P,S) = \left| \left\{ i \in [n] \;\middle|\; P^{(<s)}(\alpha_i) = \beta_i^{(<s)} \right\} \right| .$$

Theorem 4.1. *Let r be an integer with $0 \le r < s$ and $r < d/2$.*

Let
$$A_0 = \left(n \cdot \frac{\prod_{j=0}^{r}(d-j)}{\prod_{j=0}^{r}(s-j)} \right)^{\frac{1}{r+2}} .$$
Let $\varepsilon < 1$ be a real number such that
$$\varepsilon > \frac{8s^2A_0}{dq} \quad \textit{and} \quad \varepsilon > \frac{4s^2A_0}{q^2} .$$
Let $A = A_0 \cdot (1+\varepsilon)$.

Let $\mathcal{L}$ be the set of polynomials that have more than A points of agreement with S:
$$\mathcal{L} = \{ P(T) \in \mathbb{F}_q[T] \mid \deg(P) \le d, \mathrm{agree}(P,S) > A \} .$$
Then
$$|\mathcal{L}| \le (r+1) \cdot q^{O(r\frac{d}{q}+r+1)} ,$$
and there is an algorithm that can compute $\mathcal{L}$ in time $q^{O(rd/q+r+1)}$.

This theorem will be derived from the following two theorems. The first one finds a differential equation for which $P(T)$ is a solution.

Theorem 4.2. *There exists a polynomial $Q(X, Y_0, \ldots, Y_r) \in \mathbb{F}_q[X, Y_0, \ldots, Y_r]$, such that:*

1. $\deg_{Y_j}(Q) \le \frac{4As^2}{\varepsilon d}$ *for each j.*
2. *The $(1, d, d-1, \ldots, d-r)$-weighted degree of Q is $\le A \cdot \frac{2s^2}{\varepsilon}$.*
3. *For every $P(T) \in \mathcal{L}$, we have*
$$Q(T, P(T), \ldots, P^{(r)}(T)) = 0 .$$

Furthermore, Q can be found in time $\mathrm{poly}\left(n \cdot (s/\varepsilon)^r \cdot \log q\right)$.

The next theorem shows how to find all solutions to such differential equations.

Theorem 4.3. *Let $t > 0$ be an integer.*

Suppose $Q(X,Y_0,\dots,Y_r) \in \mathbb{F}_q[X,Y_0,\dots,Y_r]$ is such that:

1. $\deg_{Y_j}(Q) \le t < q$ *for each j.*

2. *The $(1,d,d-1,\dots,d-r)$-weighted degree of Q is $< q^2$.*

Then the set of all $P(T) \in \mathbb{F}_q[T]$ with degree at most d such that

$$Q(T,P(T),P^{(1)}(T),\dots,P^{(r)}(T)) = 0$$

has cardinality at most $t \cdot (r+1) \cdot q^{2r\cdot\lfloor d/q\rfloor+4r+4}$.

Furthermore, this set can be found in time $q^{O(r+\frac{rd}{q}+1)}$.

Combining these two theorems, we complete the proof of Theorem 4.1.

Proof of Theorem 4.1. We first apply Theorem 4.2 to obtain polynomial $Q(X,Y_0,\dots,Y_r)$.

This Q will satisfy the hypothesis of Theorem 4.3 (with $t = q-1$) if we have

$$\frac{4As^2}{\varepsilon d} < q \quad \text{and} \quad A \cdot \frac{2s^2}{\varepsilon} < q^2 .$$

Our conditions on ε and the fact that $A < 2A_0$ imply precisely these conditions.

Thus we may apply Theorem 4.3, and get the desired list $\mathcal{L}$. The running time of the algorithm and the list size bound follow immediately. □

We may now complete the proof of the main theorem on list-decoding univariate multiplicity codes.

Proof of Theorem 3.1. We have $\mathcal{C} \subseteq \Sigma^{\mathbb{F}_q}$, where $\Sigma = \mathbb{F}_q^s$. Suppose we are given a received word $w : \mathbb{F}_q \to \Sigma$. We think of w as a tuple of functions $w = (w^{(0)}, w^{(1)},\dots,w^{(s-1)}) : \mathbb{F}_q \to \mathbb{F}_q^s$.

Now let

$$S = \left\{ \left(\alpha, w^{(0)}(\alpha),\dots,w^{(s-1)}(\alpha)\right) \,\middle|\, \alpha \in \mathbb{F}_q \right\}.$$

We want to apply Theorem 4.1 to this set S.

We have $n = |S| = q$. We may choose

$$\varepsilon = 2 \cdot \max\left(\frac{8s^2A}{qd}, \frac{4s^2A}{q^2}\right);$$

then

$$\varepsilon > \frac{8s^2A}{dq}, \quad \varepsilon > \frac{4s^2A}{q^2}, \quad \text{and} \quad \varepsilon < 1 .$$

Then by Theorem 4.1, in time $q^{O(rs)}$, we can find all codewords $c \in \mathcal{C}$ such that $\Delta(c,w)$ is at most:

$$1 - \frac{A}{q} \le 1 - (1+\varepsilon)\frac{\left(q \cdot \left(\frac{d}{s-r}\right)^{r+1}\right)^{\frac{1}{r+2}}}{q},$$

with a list size at most $q^{O(rd/q+r+1)}$.

Finally we observe that the list decoding radius above is at least as big as our target list decoding radius:

$$1-\left(\frac{s}{s-r}\cdot R\right)^{\frac{r+1}{r+2}}-\max\left(\frac{16\cdot s^2}{d},\frac{8s^2}{q}\right),$$

(this uses the fact that $\varepsilon \le \max(16\cdot s^2/d, 8s^2/q)$ which follows from the inequalities $A \le n \le q$).

This completes the proof of the theorem. □

4.1 Finding a differential equation

Proof of Theorem 4.2. Choose parameters

$$B=(r+1)\left(s-\frac{r}{2}\right)+1,\quad M=\left\lceil\frac{2B}{\varepsilon}\right\rceil,\quad\text{and}\quad D=\lfloor AM\rfloor-1\,.$$

Observe that we have:

$$\frac{D}{M}<A\,, \tag{4.1}$$

$$\frac{D}{M+B}>A_0\,. \tag{4.2}$$

The polynomial Q will be chosen to satisfy some "weighted-multiplicity" vanishing conditions. We now give these conditions.

Fix $i\in[n]$.

Define

$$R_i(T)=\sum_{j=0}^{s-1}\beta_i^{(j)}(T-\alpha_i)^j\,.$$

Notice that if $P(T)\in\mathbb{F}_q[T]$ is such that $P^{(<s)}(\alpha_i)=\beta_i^{(<s)}$, then $P(T)=R_i(T) \mod (T-\alpha_i)^s$. Furthermore, in this case we have for each $0\le j<s$:

$$P^{(j)}(T)=R_i^{(j)}(T)\mod (T-\alpha_i)^{s-j}. \tag{4.3}$$

Let $\mathcal{M}_{i,e,e_0,\dots,e_r}(X,Y_0,\dots,Y_r)\in\mathbb{F}_q[X,Y_0,\dots,Y_r]$ be the polynomial:

$$\mathcal{M}_{i,e,e_0,\dots,e_r}(X,Y_0,\dots,Y_r)=(X-\alpha_i)^e\prod_{j=0}^{r}(Y_j-R_i^{(j)}(X))^{e_j}\,.$$

Condition C_i: The polynomial $Q(X,Y_0,\dots,Y_r)$ lies in the ideal $\mathcal{I}_i$ of $\mathbb{F}_q[X,Y_0,\dots,Y_r]$ generated by

$$\left\{\mathcal{M}_{i,e,e_0,\dots,e_r}\;\middle|\; e+\sum_{j=0}^{r}((s-j)\cdot e_j)=M\right\}.$$

Observe that every polynomial in $\mathbb{F}_q[X, Y_0, \ldots, Y_r]$ can be uniquely represented as an $\mathbb{F}_q$-linear combination of the polynomials:

$$\{\mathcal{M}_{i,e,e_0,\ldots,e_r} \mid e, e_0, \ldots, e_r \geq 0\}.$$

Asking that $Q(X, Y_0, \ldots, Y_r)$ lies in $\mathcal{I}_i$ is equivalent to asking that for each $(e, e_0, \ldots, e_r)$ such that $e + \sum_{j=0}^{r}((s-j)\cdot e_j) < M$, the coefficient of $\mathcal{M}_{e,e_0,\ldots,e_r}$ in this unique representation equals 0. Thus, this condition C_i imposes at most

$$\left|\left\{(e, e_0, \ldots, e_r) \;\middle|\; e + \sum_{j=0}^{r}((s-j)\cdot e_j) < M\right\}\right|$$

independent $\mathbb{F}_q$-linear constraints on the coefficients of Q.

Letting i vary in $[n]$, we get that the total number of independent $\mathbb{F}_q$-linear constraints imposed on the coefficients of Q is at most

$$n \cdot \left|\left\{(e, e_0, \ldots, e_r) \;\middle|\; e + \sum_{j=0}^{r}((s-j)\cdot e_j) < M\right\}\right| \leq n \cdot \frac{(M+B)^{r+2}}{(r+2)!\prod_{j=0}^{r}(s-j)},$$

(this bound is obtained by bounding the number of unit cubes that can be contained in the appropriate simplex, via a volume argument).

If the dimension of the $\mathbb{F}_q$-linear space of polynomials $Q(X, Y_0, \ldots, Y_r)$ of $(1, d, d-1, \ldots, d-r)$-weighted degree at most D is larger than the number of independent $\mathbb{F}_q$-linear constraints imposed by the C_i, then such a polynomial satisfying all the conditions C_i will exist. Counting monomials, the dimension in question is at least

$$\frac{D^{r+2}}{(r+2)!\cdot\prod_{j=0}^{r}(d-j)},$$

(again, this bound is obtained by considering the volume of a suitable simplex). Thus, such a polynomial satisfying all the C_i exists if the following condition holds:

$$\frac{D^{r+2}}{(r+2)!\cdot\prod_{j=0}^{r}(d-j)} > n \cdot \frac{(M+B)^{r+2}}{(r+2)!\prod_{j=0}^{r}(s-j)},$$

which follows from

$$\frac{D}{M+B} > \left(n \cdot \frac{\prod_{j=0}^{r}(d-j)}{\prod_{j=0}^{r}(s-j)}\right)^{\frac{1}{r+2}} = A_0,$$

and we know this (4.2).

Furthermore, this polynomial Q can be found in time

$$\mathrm{poly}\left(\frac{D^{r+2}}{(r+2)!\cdot\prod_{j=0}^{r}(d-j)} \cdot \log q\right) = \mathrm{poly}\left(n \cdot \left(\frac{s}{\varepsilon}\right)^{r} \cdot \log q\right)$$

by solving the system of linear equations.

We now show that Q has the desired properties.

The $(1,d,d-1,\dots,d-r)$ weighted degree of Q is at most

$$D < AM \le A\cdot\left(\frac{2B}{\varepsilon}+1\right) \le A\cdot\frac{2s^2}{\varepsilon}\,.$$

Because of this, we have that the

$$\deg_{Y_j}(Q) \le \frac{D}{d-j} \le \frac{D}{d-r} \le \frac{2D}{d} \le A\cdot\frac{4s^2}{\varepsilon d}\,.$$

Now fix any $P(T)\in\mathcal{L}$. Let $\mathcal{A}$ be the set of agreement points:

$$\mathcal{A} = \left\{ i\in[n] \;\middle|\; P^{(<s)}(\alpha_i) = \beta_i^{(<s)} \right\}.$$

Consider the polynomial $H(T) = Q(T,P(T),P^{(1)}(T),\dots,P^{(r)}(T))$. Notice that $\deg(H)\le D$. We now show that for each point of agreement $i\in\mathcal{A}$, $H(T)$ vanishes at α_i with multiplicity at least M. Using the fact that $Q(X,Y_0,\dots,Y_r)$ lies in $\mathcal{I}_i$, we have that for every $i\in\mathcal{A}$, $H(T)$ can be written in the form

$$Q(T,P(T),\dots,P^{(s-1)}(T)) = \sum_{e,e_0,\dots,e_r \mid e+\sum((s-j)e_j)=M} B_{i,e,e_0,\dots,e_r}(T)\mathcal{M}_{i,e,e_0,\dots,e_r}(T,P(T),\dots,P^{(r)}(T))\,.$$

Recall that $\mathcal{M}_{i,e,e_0,\dots,e_r}(T,P(T),\dots,P^{(r)}(T))$ equals

$$(T-\alpha_i)^e\prod_{j=0}^{r}\left(P^{(j)}(T)-R_i^{(j)}(T)\right)^{e_i},$$

which for $i\in\mathcal{A}$, is divisible by $(T-\alpha)^{e+\sum((s-j)\cdot e_j)}$ (because of (4.3)). Thus for each $i\in\mathcal{A}$, $H(T)$ vanishes at α_i with multiplicity at least M.

Thus $H(T)$ vanishes with multiplicity at least M on $|\mathcal{A}|$ points. Since $M\cdot|\mathcal{A}|\ge M\cdot A > D$ (by (4.1)), we conclude that $H(T)=0$.

This completes the proof of the theorem. □

4.2 Solving the differential equation

We now prove Theorem 4.3. The main idea is to consider the power series expansion of the potential solution. If a suitable nonsingularity condition holds, this approach will succeed. This is the content of the following theorem. Later we will see that if the nonsingularity condition does not hold, we will be able to make progress by solving a related differential equation.

In our algorithm, we may have to work with power series over the extension field $\mathbb{F}_{q^2}$ of $\mathbb{F}_q$ (this allows us to try out enough "initial conditions"; and $\mathbb{F}_q$ turns out to be too small to allow this). We therefore state the next theorem over a general field $\mathbb{F}$.

Theorem 4.4. *Let $\mathbb{F}$ be a field of characteristic p.*

Let $Q(X,Y_0,\dots,Y_r)\in\mathbb{F}[X,Y_0,\dots,Y_r]$. Let $k\ge 1$ be an integer such that p does not divide $\binom{k+r}{r}$. Suppose $f(T)\in\mathbb{F}[T]$ and $\alpha\in\mathbb{F}$ are such that:

1. $\dfrac{\partial Q}{\partial Y_r}(\alpha, f(\alpha), f^{(1)}(\alpha), \ldots, f^{(r)}(\alpha)) \neq 0$,

2. $Q(T, f(T), f^{(1)}(T), \ldots, f^{(r)}(T)) = 0 \mod (T-\alpha)^k$.

Then there exists a unique $\gamma \in \mathbb{F}$ such that the function $g(T) = f(T) + \gamma \cdot (T-\alpha)^{k+r}$ satisfies:

$$Q(T, g(T), g^{(1)}(T), \ldots, g^{(r)}(T)) = 0 \mod (T-\alpha)^{k+1}.$$

Furthermore, this γ can be found efficiently.

Proof. Since

$$Q(T, f(T), f^{(1)}(T), \ldots, f^{(r)}(T)) = 0 \mod (T-\alpha)^k,$$

we know that

$$Q(T, f(T), f^{(1)}(T), \ldots, f^{(r)}(T)) = \beta \cdot (T-\alpha)^k \mod (T-\alpha)^{k+1}$$

for some $\beta \in \mathbb{F}$.

Let $g(T) = f(T) + \gamma \cdot (T-\alpha)^{k+r}$. Then for each $i \leq r$, we have that

$$g^{(i)}(T) = f^{(i)}(T) + \gamma \cdot \binom{k+r}{i} (T-\alpha)^{k+r-i}.$$

In particular, we also have $g^{(i)}(T) = f^{(i)}(T) \mod (T-\alpha)^k$.

Thus by Taylor expansion, we can write $Q(T, g(T), g^{(1)}(T), \ldots, g^{(r)}(T))$ as:

$$Q(T, f(T), \ldots, f^{(r)}(T)) + \sum_{i=0}^{r} \frac{\partial Q}{\partial Y_i}(T, f(T), \ldots, f^{(r)}(T)) \cdot \gamma \binom{k+r}{i} (T-\alpha)^{k+r-i} + (T-\alpha)^{2k} \cdot h(T),$$

for some $h(T) \in \mathbb{F}[T]$ (the last term in the above expression arises from the degree-2 and higher terms in the Taylor expansion; since each $g^{(i)}(T) - f^{(i)}(T)$ is divisible by $(T-\alpha)^k$, we get that all the degree-2 and higher terms in the Taylor expansion are divisible by $(T-\alpha)^{2k}$).

Considering this equation mod $(T-\alpha)^{k+1}$ (and using the fact that $2k \geq k+1$), we get:

$$\begin{aligned} &Q(T, g(T), g^{(1)}(T), \ldots, g^{(r)}(T)) \\ &\quad = (T-\alpha)^k \cdot \left(\beta + \frac{\partial Q}{\partial Y_r}(T, f(T), \ldots, f^{(r)}(T)) \cdot \gamma \binom{k+r}{r}\right) \mod (T-\alpha)^{k+1} \\ &\quad = (T-\alpha)^k \cdot \left(\beta + \frac{\partial Q}{\partial Y_r}(\alpha, f(\alpha), \ldots, f^{(r)}(\alpha)) \cdot \gamma \binom{k+r}{r}\right) \mod (T-\alpha)^{k+1}. \end{aligned}$$

Note that

$$\frac{\partial Q}{\partial Y_r}(\alpha, f(\alpha), \ldots, f^{(r)}(\alpha)) \cdot \binom{k+r}{r}$$

is nonzero in $\mathbb{F}$ (by the hypotheses of the theorem), and thus there is unique γ that makes $Q(T, g(T), g^{(1)}(T), \ldots, g^{(r)}(T))$ equal 0 mod $(T-\alpha)^{k+1}$, namely:

$$\gamma = -\frac{\beta}{\frac{\partial Q}{\partial Y_r}(\alpha, f(\alpha), \ldots, f^{(r)}(\alpha)) \cdot \binom{k+r}{r}}. \qquad \square$$

This theorem lets us start with an "initial solution" of the differential equation mod $(T-\alpha)$, and successively improve it to find solutions mod $(T-\alpha)^2$, $(T-\alpha)^3$, etc., till we get all solutions of degree d. We record this below.

Corollary 4.5. *Let $\mathbb{F}$ be a finite field of characteristic p. Let $Q(X,Y_0,\dots,Y_r) \in \mathbb{F}[X,Y_0,\dots,Y_r]$. Let $\alpha,\beta^{(0)},\beta^{(1)},\dots,\beta^{(r)} \in \mathbb{F}$. Suppose that*

$$\frac{\partial Q}{\partial Y_r}(\alpha,\beta^{(0)},\dots,\beta^{(r)}) \neq 0\,.$$

Let $\mathcal{L}$ be the set of all polynomials $P(T) \in \mathbb{F}[T]$ of degree at most d satisfying

$$Q(T,P(T),P^{(1)}(T),\dots,P^{(r)}(T)) = 0\,,$$

and the initial conditions:

$$P(\alpha) = \beta^{(0)}, P^{(1)}(\alpha) = \beta^{(1)},\dots,P^{(r)}(\alpha) = \beta^{(r)}\,.$$

Then $|\mathcal{L}| \le |\mathbb{F}|^{r\cdot\lfloor d/p\rfloor+r}$, and $\mathcal{L}$ can be found in time $|\mathbb{F}|^{O(r\cdot\lfloor d/p\rfloor+r)}$.

Proof. We describe an algorithm to find $\mathcal{L}$. From the description of the algorithm, the bound on $|\mathcal{L}|$ will be clear.

Start with $f_0(T) = \beta^{(0)} + \beta^{(1)}(T-\alpha) + \cdots + \beta^{(r)}(T-\alpha)^r$.

1. Let $\mathcal{L}_r = \{f_0(T)\}$.
2. For $i = r+1, r+2, \dots, d$, do:
   (a) If $\binom{i}{r}$ is relatively prime to p,
   - For each $f(T) \in \mathcal{L}_{i-1}$, let $g(T) = f(T) + \gamma\cdot(T-\alpha)^i$ be what is given by Theorem 4.4, and include $g(T)$ into $\mathcal{L}_i$.

   (b) Otherwise,
   - For each $f(T) \in \mathcal{L}_{i-1}$ and for each $\gamma \in \mathbb{F}$, let $g(T) = f(T) + \gamma\cdot(T-\alpha)^i$, and include $g(T)$ into $\mathcal{L}_i$.
3. Let $\mathcal{L}^* = \left\{P(T) \in \mathcal{L}_d \mid Q(T,P(T),P^{(1)}(T),\dots,P^{(r)}(T)) = 0\right\}$.
4. Output $\mathcal{L}^*$.

The correctness of this algorithm follows easily from Theorem 4.4. The invariant maintained is that

$$\mathcal{L}_i \supseteq \{P(T) \bmod (T-\alpha)^{i+1} \mid P(T) \in \mathcal{L}\}\,.$$

To understand $|\mathcal{L}|$, we observe that $|\mathcal{L}| = |\mathcal{L}^*| \le |\mathcal{L}_d|$, $|\mathcal{L}_r| = 1$, and for each $i \le d$, we have $|\mathcal{L}_i| = |\mathcal{L}_{i-1}|$ if $\binom{i}{r}$ is relatively prime to p, and $|\mathcal{L}_i| = |\mathbb{F}|\cdot|\mathcal{L}_{i-1}|$ otherwise.

This implies that $|\mathcal{L}| \le |\mathbb{F}|^c$, where c is the number of integers i in $\{r+1,\dots,d\}$ such that $\binom{i}{r}$ is divisible by p. This implies the claimed bounds on the running time of the algorithm and the size of the output list. □

We can now prove the main theorem on the solvability of polynomial differential equations, Theorem 4.3.

Proof of Theorem 4.3. We give the algorithm below, the bound on the number of solutions will follow from the analysis of the algorithm.

Algorithm $\mathsf{SOLVE}(Q)$**:**
Here $Q(X,Y_0,\ldots,Y_r) \in \mathbb{F}_q[X,Y_0,\ldots,Y_r]$ is a nonzero polynomial with $\deg_{Y_i}(Q) < q$ for each i.

1. Initialize $\mathcal{L} = \emptyset$.

2. If $Q(X,Y_0,\ldots,Y_r)$ does not depend on any of $Y_0,\ldots,Y_r$, return $\mathcal{L}$ and exit.

3. Let r^* be the largest j such that $Q(X,Y_0,\ldots,Y_r)$ depends on Y_j.

4. For each $(\alpha,\beta_0,\ldots,\beta_{r^*}) \in \mathbb{F}_{q^2}^{r^*+2}$, do the following:

   - If
   $$Q(\alpha,\beta_0,\ldots,\beta_{r^*}) = 0 \quad \text{and} \quad \frac{\partial Q}{\partial Y_{r^*}}(\alpha,\beta_0,\ldots,\beta_{r^*}) \neq 0,$$
   then using the algorithm of Corollary 4.5 (with $\mathbb{F} = \mathbb{F}_{q^2}$), find all[2] $P(T) \in \mathbb{F}_q[T]$ of degree at most d with $P^{(j)}(\alpha) = \beta_j$ for $0 \le j \le r^*$, satisfying $Q(T,P(T),P^{(1)}(T),\ldots,P^{(r^*)}(T)) = 0$.
   - Include all such $P(T)$ in the list $\mathcal{L}$.

5. Let $Q^{\#}(X,Y_0,\ldots,Y_{r^*}) = \dfrac{\partial Q}{\partial Y_{r^*}}(X,Y_0,\ldots,Y_{r^*})$.

6. Let $\mathcal{L}^{\#} = \mathsf{SOLVE}(Q^{\#})$.

7. Output $\mathcal{L} \cup \mathcal{L}^{\#}$.

Correctness. We will now prove correctness of the algorithm. Namely, every solution of degree at most d of the polynomial-differential equation

$$Q(T,P(T),P^{(1)}(T),\ldots,P^{(r)}(T)) = 0$$

is included in the output of the algorithm.

Let r^* be the largest j such that $Q(X,Y_0,\ldots,Y_r)$ depends on Y_j. We prove the correctness of the algorithm by induction on r^* and $\deg_{Y_{r^*}}(Q)$.

The base case is when Q does not depend on any of $Y_0,\ldots,Y_r$. In this case, the algorithm outputs $\emptyset$, which is correct.

Now suppose we know correctness of the algorithm for all polynomials Q which depend on only (some subset of) $Y_0,\ldots,Y_{r^*-1}$, and for all polynomials Q which depend on only (some subset of) $Y_0,\ldots,Y_{r^*}$ with $\deg_{Y_{r^*}}(Q) < \mathcal{D}$.

[2]More precisely, we use the algorithm of Corollary 4.5 to find all such $P(T) \in \mathbb{F}_{q^2}[T]$, and then we only keep the ones that are in $\mathbb{F}_q[T]$.

We now show that the algorithm is correct for all polynomials Q which depend on (some subset of) $Y_0,\dots,Y_{r^*}$ and with $\deg_{Y_{r^*}}(Q) \le \mathcal{D}$. Suppose $\hat{P}(T) \in \mathbb{F}_q[T]$ is a polynomial of degree at most d such that

$$Q(T,\hat{P}(T),\hat{P}^{(1)}(T),\dots,\hat{P}^{(r^*)}(T)) = 0\,.$$

Case 1: Suppose $\hat{P}(T)$ also satisfies the differential equation

$$\frac{\partial Q}{\partial Y_{r^*}}(T,\hat{P}(T),\hat{P}^{(1)}(T),\dots,\hat{P}^{(r^*)}(T)) = 0\,.$$

Note that since $0 < \deg_{Y_{r^*}}(Q) < q$, the polynomial

$$\frac{\partial Q}{\partial Y_{r^*}}(X,Y_0,\dots,Y_r)$$

is *nonzero*. Then by the induction hypothesis applied to the polynomial

$$\frac{\partial Q}{\partial Y_{r^*}}(X,Y_0,\dots,Y_r)\,,$$

$\hat{P}(T)$ will be included in the list $\mathcal{L}^{\#}$, and hence in the output of the algorithm.

Case 2: Otherwise, we know that $\hat{P}(T)$ is such that:

$$H(T) = \frac{\partial Q}{\partial Y_{r^*}}(T,\hat{P}(T),\hat{P}^{(1)}(T),\dots,\hat{P}^{(r^*)}(T)) \neq 0\,.$$

Note that the degree of $H(T)$ is less than q^2.

Thus there is some $\alpha \in \mathbb{F}_{q^2}$ such that $H(\alpha) \neq 0$. For that α, we have

$$Q(\alpha,\hat{P}(\alpha),\hat{P}^{(1)}(\alpha),\dots,\hat{P}^{(r^*)}(\alpha)) \neq 0\,.$$

Thus in the iteration of Step 4 of the algorithm in which we take:

$$\beta^{(0)} = \hat{P}(\alpha), \quad \beta^{(1)} = \hat{P}^{(1)}(\alpha), \qquad \dots \qquad \beta^{(r^*)} = \hat{P}^{(r^*)}(\alpha)\,,$$

$\hat{P}(T)$ will be included in the list $\mathcal{L}$, and hence in the output of the algorithm.

Thus in either case $\hat{P}(T)$ appears in the output of the algorithm, as desired.

Running time. The running time of Step 4 of the algorithm can be bounded by

$$q^{2(r+2)} \cdot q^{O(\frac{d}{q}\cdot r + r)} = q^{O(r+\frac{dr}{q})}\,.$$

The total number of recursive calls of the algorithm is bounded by

$$\sum_{j=0}^{r} (\deg_{Y_j}(Q)+1) \le (r+1)\cdot q\,.$$

Thus the total running time of the algorithm can be bounded by $q^{O(r+\frac{dr}{q})}$.

List size. The total number of solutions to the differential equation found by Step 4 is bounded by

$$q^{2(r+2)+2r\cdot\lfloor\frac{d}{q}\rfloor+2r}.$$

Unravelling the recursion, we get that the total number of solutions found by the algorithm is at most

$$(r+1)\cdot t\cdot q^{2(r+2)+2r\cdot\lfloor\frac{d}{q}\rfloor+2r}.$$ □

5 List-decoding multivariate multiplicity codes

In this section, we will prove Theorem 3.2 on the list-decoding of multivariate multiplicity codes over $\mathbb{F}_q^m$.

The main technical result is the following, which reduces the problem of list-decoding of multivariate multiplicity codes over $\mathbb{F}_q^m$ to several instances of list-decoding univariate multiplicity codes over $\mathbb{F}_{q^m}$.

Theorem 5.1. *Suppose there exists an algorithm $\mathcal{A}$ which runs in time $\mathcal{T}$ and list-decodes the multiplicity code of order-s evaluations of degree-dq^{m-1} polynomials in 1 variable over $\mathbb{F}_{q^m}$ from an η fraction of errors with list-size at most L.*

Then there exists an algorithm that list-decodes the multiplicity code of order-s evaluations of degree-d polynomials in m variables over $\mathbb{F}_q$ from an η fraction of errors, which runs in time

$$\mathcal{T}\cdot\binom{m+s-1}{m}+\mathrm{poly}\left(q^m,\binom{d+m}{m}\right)\cdot L^{\binom{m+s-1}{m}},$$

and has list-size at most $L^{\binom{m+s-1}{m}}$.

Given this, we now complete the proof of Theorem 3.2.

Proof of Theorem 3.2. The main input that we need is a good list-decoding algorithm for univariate multiplicity codes. Such an algorithm follows from [11]. In particular, this says that every order-s univariate multiplicity code over $\mathbb{F}_q$ with distance δ can be list-decoded from a $(1-\sqrt{1-\delta})$ fraction of errors with list-size $O(q)$ in time $\mathrm{poly}(q^s)$, and from a $(1-\sqrt{1-\delta}-\varepsilon)$ fraction of errors with list-size $\mathrm{poly}(\frac{1}{\varepsilon})$ in time $\mathrm{poly}(q^s)$. (The same algorithm, in more a explicit form, is given by the $r=0$ case of Theorem 3.1, but the analysis in Theorem 4.3 has to be done more carefully to get the claimed list-size bound.)

Combining this list-decoding algorithm for univariate multiplicity codes with Theorem 5.1, we get the desired list-decoding algorithm. Explicitly, it gives an algorithm that list-decodes m-variate order-s multiplicity codes of degree-d polynomials over $\mathbb{F}_q$ with the following two sets of parameters:

1.
   - from a $\left(1-\sqrt{1-\frac{d\cdot q^{m-1}}{s\cdot q^m}}\right)$ fraction of errors,
   - with list-size $q^{(m+O(1))\cdot\binom{m+s-1}{m}}$,
   - in time $\mathrm{poly}(q^m)\cdot\binom{m+s-1}{m}+\mathrm{poly}(q^m)\cdot q^{m\cdot\binom{m+s-1}{m}}$.

2. • from a $\left(1-\sqrt{1-\frac{d\cdot q^{m-1}}{s\cdot q^m}}-\varepsilon\right)$ fraction of errors,
   • with list-size $\left(\frac{1}{\varepsilon}^{O\left(\binom{m+s-1}{m}\right)}\right)$,
   • in time $\operatorname{poly}(q^m)\cdot\binom{m+s-1}{m}+\operatorname{poly}(q^m)\cdot\left(\frac{1}{\varepsilon}\right)^{O\left(\binom{m+s-1}{m}\right)}$.

Noting that

$$\frac{d\cdot q^{m-1}}{s\cdot q^m}=\delta\,,$$

this completes the proof of the theorem. □

The rest of this section is devoted to a proof of Theorem 5.1.

5.1 $\mathbb{F}_{q^m}$ and $\mathbb{F}_q^m$

Let $\mathrm{Tr}:\mathbb{F}_{q^m}\to\mathbb{F}_q$ denote the finite field trace:

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

We also use Tr to denote the associated polynomial:

$$\mathrm{Tr}(T)=T+T^q+\cdots+T^{q^{m-1}}\in\mathbb{F}_{q^m}[T]\,.$$

We will be working with bases for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Abusing notation, we will say that

$$\mathfrak{a}=(\alpha_1,\alpha_2,\ldots,\alpha_m)\in\mathbb{F}_{q^m}^m$$

is a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ if $\{\alpha_1,\alpha_2,\ldots,\alpha_m\}$ is a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

Definition 5.2 (s-general position)**.** Let $\mathfrak{a}_1,\mathfrak{a}_2,\ldots,\mathfrak{a}_M\in\mathbb{F}_{q^m}^m$ be bases for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We say that $\mathfrak{a}_1,\ldots,\mathfrak{a}_M$ are in *s-general position* if for every $R(X_1,\ldots,X_m)\in\mathbb{F}_{q^m}[X_1,\ldots,X_m]$ with $\deg(R)<s$, there is some $j\in[M]$ such that $R(\mathfrak{a}_j)\neq 0$.

Lemma 5.3. *If $s\le q^m-q^{m-1}$, and $M\ge\binom{m+s-1}{m}$, there exists a collection of bases $\mathfrak{a}_1,\ldots,\mathfrak{a}_M\in\mathbb{F}_{q^m}^m$ in s-general position. Furthermore, such a collection can be found in time*

$$\operatorname{poly}\left((q^m)^{\binom{m+s-1}{m}}\right)\,.$$

Proof. Note that the property of being in s-general position is monotone, and so it will suffice to prove the lemma with $M=\binom{m+s-1}{m}$.

Let $\mathcal{B}\subseteq\mathbb{F}_{q^m}^m$ be the set of all bases of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. It is a well known fact that $\mathcal{B}$ can be described as the set of non-zeroes of a certain degree-q^{m-1} polynomial as follows:

$$\mathcal{B}=\{(x_1,\ldots,x_m)\in\mathbb{F}_{q^m}^m\mid Z(x_1,\ldots,x_m)\neq 0\}\,,$$

where:

$$Z(X_1,\dots,X_m)=\det\begin{pmatrix} X_1 & X_2 & \dots & X_m \\ X_1^q & X_2^q & \dots & X_m^q \\ X_1^{q^2} & X_2^{q^2} & \dots & X_m^{q^2} \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{q^{m-1}} & X_2^{q^{m-1}} & \dots & X_m^{q^{m-1}} \end{pmatrix}.$$

The space of all $R(X_1,\dots,X_m)\in\mathbb{F}_{q^m}[X_1,\dots,X_m]$ with $\deg(R)<s$ is $\binom{m+s-1}{m}$-dimensional. To each such polynomial R, we associate a vector $v_R\in\mathbb{F}_{q^m}^{\mathcal{B}}$, where the coordinate of v_R corresponding to $(x_1,\dots x_m)\in\mathcal{B}$ equals $R(x_1,\dots,x_m)$.

Now we will show that the dimension of

$$\{v_R \mid R(X_1,\dots,X_m)\in\mathbb{F}_{q^m}[X_1,\dots,X_m],\deg(R)\le s\}$$

also equals $\binom{m+s-1}{m}$. Otherwise, there should exist some nonzero polynomial $R(X_1,\dots,X_m)$ such that $R(x_1,\dots,x_m)=0$ for every $(x_1,\dots,x_m)\in\mathcal{B}$. Thus the polynomial $R\cdot Z(X_1,\dots,X_m)$ would vanish on every $(x_1,\dots,x_m)\in\mathbb{F}_{q^m}^m$, which is impossible since $\deg(R\cdot Z)<s+q^{m-1}\le q^m$.

Therefore, there is a subset

$$A=\left\{\mathfrak{a}_1,\dots,\mathfrak{a}_{\binom{m+s-1}{m}}\right\}$$

of $\mathcal{B}$ of size $\binom{m+s-1}{m}$ such that the set of restrictions of the vectors v_R to coordinates in A also has dimension $\binom{m+s-1}{m}$. This implies that for every nonzero R, the R must be nonzero on at least one of $\mathfrak{a}_1,\dots,\mathfrak{a}_{\binom{m+s-1}{m}}$.

This argument can be made an algorithm for finding the $\mathfrak{a}_i$ by considering the matrix whose rows consist of the v_R, and then finding a maximal collection of linearly independent columns of it. □

Explicit bases in s-general position. Though not required for the list-decoding application, we remark that it is possible to construct bases in s-general position in time $\mathrm{poly}(q,m)$, provided $s<q$. Let $S\subseteq\mathbb{F}_q^m$ be an *interpolating set* for degree $<s$ polynomials (there are many known simple constructions of such sets). Let $\mathfrak{a}\in\mathbb{F}_{q^m}^m$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then $\{v+\mathfrak{a}\mid v\in S\}\subseteq\mathbb{F}_{q^m}^m$ is a set of bases in s-general position.

5.2 Curves parametrizing $\mathbb{F}_q^m$

We will be interested in certain curves parametrizing $\mathbb{F}_q^m$. Associated to every basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, there is one such curve. Let $\mathfrak{a}=(\alpha_1,\dots,\alpha_m)\in\mathbb{F}_{q^m}^m$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Define the curve $\gamma_{\mathfrak{a}}:\mathbb{F}_{q^m}\to\mathbb{F}_q^m$ given by

$$\gamma_{\mathfrak{a}}(t)=(\mathrm{Tr}(\alpha_1 t),\mathrm{Tr}(\alpha_2 t),\dots,\mathrm{Tr}(\alpha_m t)).$$

We also view $\gamma_{\mathfrak{a}}$ as tuple of polynomials:

$$\gamma_{\mathfrak{a}}(T)=(\mathrm{Tr}(\alpha_1 T),\mathrm{Tr}(\alpha_2 T),\dots,\mathrm{Tr}(\alpha_m T))\in(\mathbb{F}_{q^m}[T])^m.$$

We now list the salient features of the curve $\gamma_{\mathfrak{a}}$.

Lemma 5.4 (Properties of $\gamma_{\mathfrak{a}}$)**.** *Let $\mathfrak{a}$, $\gamma_{\mathfrak{a}}$ be as above. Then:*

- $\gamma_{\mathfrak{a}}$ parametrizes $\mathbb{F}_q^m$: $\gamma_{\mathfrak{a}}(\mathbb{F}_{q^m}) = \mathbb{F}_q^m$. *In particular $\gamma_{\mathfrak{a}}$ gives a bijection between $\mathbb{F}_{q^m}$ and $\mathbb{F}_q^m$.*
- $\gamma_{\mathfrak{a}}$ is $\mathbb{F}_q$-linear: *For $\alpha_1, \alpha_2 \in \mathbb{F}_q$, we have*
$$\gamma_{\mathfrak{a}}(\alpha_1 T_1 + \alpha_2 T_2) = \alpha_1 \gamma_{\mathfrak{a}}(T_1) + \alpha_2 \gamma_{\mathfrak{a}}(T_2)\,.$$
- $\gamma_{\mathfrak{a}}$ looks linear mod T^q: *In the ring $\mathbb{F}_{q^m}[T]$, we have $\gamma_{\mathfrak{a}}(T) \equiv \mathfrak{a}T \mod T^q$.*

This lemma follows from elementary properties of the Tr function, see, e. g., [19].

These properties of $\gamma_{\mathfrak{a}}$ allow us to relate the evaluations of a multivariate polynomial and its derivatives over $\mathbb{F}_q^m$ to the evaluations of a univariate polynomial and its derivatives over $\mathbb{F}_{q^m}$. We do this in the following lemma.

Lemma 5.5. *Let $Q(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ and let $P(T) \in \mathbb{F}_{q^m}[T]$ be given by $P(T) = Q \circ \gamma_{\mathfrak{a}}(T)$. Then for every $t \in \mathbb{F}_{q^m}$ and every $j < q$:*
$$P^{(j)}(t) = \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}}(t))\mathfrak{a}^{\mathbf{i}}\,.$$

Proof. By definition of derivatives, we have:
$$P(t+W) = \sum_j P^{(j)}(t) W^j\,,$$
$$Q(\gamma_{\mathfrak{a}}(t) + \mathbf{X}) = \sum_{\mathbf{i}} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}}(t))\mathbf{X}^{\mathbf{i}}\,.$$

By linearity, $\gamma_{\mathfrak{a}}(t+W) = \gamma_{\mathfrak{a}}(t) + \gamma_{\mathfrak{a}}(W)$. So
$$\begin{aligned} P(t+W) &= Q \circ \gamma_{\mathfrak{a}}(t+W) \\ &= Q(\gamma_{\mathfrak{a}}(t) + \gamma_{\mathfrak{a}}(W)) \\ &= \sum_{\mathbf{i}} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}}(t))(\gamma_{\mathfrak{a}}(W))^{\mathbf{i}}\,. \end{aligned}$$

Taking this equation mod W^q, we get the following equation:
$$\sum_{j<q} P^{(j)}(t) W^j = \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})<q} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}}(t))(\mathfrak{a}W)^{\mathbf{i}} \mod W^q\,.$$

For $j < q$, note that the coefficient of W^j in the right hand side of this equation equals
$$\sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}}(t))\mathfrak{a}^{\mathbf{i}}\,.$$

On the other hand, the coefficient of W^j in the left hand side of the equation equals $P^{(j)}(t)$. We therefore conclude that for each j with $0 \le j < q$, we have
$$P^{(j)}(t) = \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}}(t))\mathfrak{a}^{\mathbf{i}}\,.$$
□

5.3 Reducing multivariate decoding to univariate decoding

Using the $\gamma_{\mathfrak{a}}$ parametrization of $\mathbb{F}_q^m$, we will now see how to reduce decoding of multivariate multiplicity codes over $\mathbb{F}_q^m$ to several instances of decoding univariate multiplicity codes over $\mathbb{F}_{q^m}$.

Proof of Theorem 5.1. We first describe the list-decoding algorithm, and then we analyze it.

Algorithm for Reducing Multivariate Decoding to Univariate Decoding:

1. Let $M = \binom{m+s-1}{m}$. Pick bases $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_M \in \mathbb{F}_{q^m}^m$ in s-general position.

2. For each $i \in [M]$, define $\ell_i : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}^s$ as follows. For each j with $0 \leq j < s$, let

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

3. Using algorithm $\mathcal{A}$, compute the set $\mathcal{L}_i$ of all $P(T) \in \mathbb{F}_{q^m}[T]$ of degree at most dq^{m-1} such that

$$\Delta(\mathrm{Enc}_{s,dq^{m-1},1,q^m}(P), \ell_i) < \eta\,.$$

4. For every

$$(P_1(T), P_2(T), \ldots, P_M(T)) \in \prod_{i=1}^{M} \mathcal{L}_i\,,$$

find all $Q(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ with $\deg(Q) \leq d$ such that for each $i \in [M]$,

$$Q \circ \gamma_{\mathfrak{a}_i}(T) = P_i(T)\,.$$

(This is a system of linear equations over $\mathbb{F}_{q^m}$ with $\binom{d+m}{m}$ variables and $(d+1) \cdot M$ constraints.)

5. Output the list of all such $Q(X_1, \ldots, X_m)$.

We first show that in step 4 of the algorithm, there can be at most one Q satisfying the given system of equations. This immediately implies that the algorithm runs in the claimed time and has the claimed list-size bound.

Suppose there were two distinct polynomials $Q_1(X_1, \ldots, X_m)$ and $Q_2(X_1, \ldots, X_m)$ which satisfied the system of linear equations in Step 4 of the algorithm. Then their difference $Q(X_1, \ldots, X_m)$ would have the property that for each $i \in [M]$, the univariate polynomial $Q \circ \gamma_{\mathfrak{a}_i}(T)$ is identically 0. By the following theorem, we get that Q must be identically 0, contradicting the distinctness of Q_1 and Q_2.

Theorem 5.6. *Let $s < q$. Let $Q(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ be a polynomial of degree $d < sq$.*
Let $\mathfrak{a}_1, \ldots, \mathfrak{a}_M \in \mathbb{F}_{q^m}^m$ be bases of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ in s-general position.
Suppose that for each $i \in [M]$, the polynomial $Q \circ \gamma_{\mathfrak{a}_i}(T) \in \mathbb{F}_{q^m}[T]$ equals 0. Then $Q(X_1, \ldots, X_m) = 0$.

We postpone the proof of this theorem to the end of this section.

Now we show correctness of the algorithm. Namely, we show that for any $\hat{Q}(X_1,\dots,X_m) \in \mathbb{F}_q[X_1,\dots,X_m]$, such that $\Delta(\mathrm{Enc}_{s,d,m,q}(\hat{Q}),r) < \eta$, the algorithm will include $\hat{Q}(X_1,\dots,X_m)$ in its output.

We say $\mathbf{a} \in \mathbb{F}_q^m$ is a *point of agreement* if $\hat{Q}^{(<s)}(\mathbf{a}) = r(\mathbf{a})$.

Fix $i \in [M]$. Let $\hat{P}_i(T) = \hat{Q} \circ \gamma_{\mathfrak{a}_i}(T)$. We will show that

$$\Delta\big(\mathrm{Enc}_{s,dq^{m-1},1,q^m}(\hat{P}_i),\ell_i\big) < \eta \,.$$

Recall that $\gamma_{\mathfrak{a}_i} : \mathbb{F}_{q^m} \to \mathbb{F}_q^m$ is a bijection. Thus for $\mathbf{a} \in \mathbb{F}_q^m$ there is a unique $t_{\mathbf{a}} \in \mathbb{F}_{q^m}$ such that $\gamma_{\mathfrak{a}_i}(t_{\mathbf{a}}) = \mathbf{a}$. Now for every point of agreement $\mathbf{a}$, we have that for every j with $0 \le j < s$:

$$\begin{aligned}
(\hat{P}_i)^{(j)}(t_{\mathbf{a}}) &= \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}_i}(t_{\mathbf{a}}))\mathfrak{a}_i^{\mathbf{i}} && \text{by Lemma 5.5,}\\
&= \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\mathbf{a})\mathfrak{a}_i^{\mathbf{i}} \\
&= \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} r^{(\mathbf{i})}(\mathbf{a})\mathfrak{a}_i^{\mathbf{i}} && \text{since } \mathbf{a} \text{ is a point of agreement,}\\
&= \sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} r^{(\mathbf{i})}(\gamma_{\mathfrak{a}_i}(t_{\mathbf{a}}))\mathfrak{a}_i^{\mathbf{i}} \\
&= (\ell_i(t_{\mathbf{a}}))_j \,.
\end{aligned}$$

Thus every point of agreement $\mathbf{a}$ gives rise to an agreement $(\hat{P}_i)^{(<s)}(t_{\mathbf{a}}) = \ell_i(t_{\mathbf{a}})$, and therefore

$$\Delta(\mathrm{Enc}_{s,dq^{m-1},1,q^m}(\hat{P}_i),\ell_i) < \eta \,.$$

Thus when we finish Step 3 of the algorithm, we know that $\hat{P}_i \in \mathcal{L}_i$ for each $i \in [M]$. Thus when we execute Step 4 of the algorithm with $P_i = \hat{P}_i$ for each $i \in [M]$, $\hat{Q}$ will be a solution to the system of linear equations (By the earlier discussion, it will in fact be the unique solution.)

Thus $\hat{Q}$ will be in the output of the algorithm, as desired. □

We now prove Theorem 5.6.

Proof of Theorem 5.6. We will show that for each $\mathbf{a} \in \mathbb{F}_q^m$, $\mathsf{mult}(Q,\mathbf{a}) \ge s$. Then by Lemma 2.3 (recalling that $\deg(Q) < sq$), we can conclude that $Q(X_1,\dots,X_m) = 0$.

Fix $i \in [M]$. We have $Q \circ \gamma_{\mathfrak{a}_i}(T) = 0$. By Lemma 5.5, we conclude that for every $t \in \mathbb{F}_{q^m}$,

$$\sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\gamma_{\mathfrak{a}_i}(t))\mathfrak{a}_i^{\mathbf{i}} = 0 \,.$$

Thus for every $\mathbf{a} \in \mathbb{F}_q^m$ and every $i \in [M]$

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

For $0 \leq j < q$, let $R_{\mathbf{a},j}(\mathbf{Y})$ be the polynomial $\sum_{\mathbf{i}:\mathrm{wt}(\mathbf{i})=j} Q^{(\mathbf{i})}(\mathbf{a})\mathbf{Y}^{\mathbf{i}}$.

We just showed that for each $i \in [M]$ and $j < q$, we have $R_{\mathbf{a},j}(\mathfrak{a}_i) = 0$.

By the general position hypothesis on $\mathfrak{a}_i$, this implies that for each $j < s$, the polynomial $R_{\mathbf{a},j}(\mathbf{Y})$ is itself identically 0.

But the coefficients of $R_{\mathbf{a},j}(\mathbf{Y})$ are $Q^{(\mathbf{i})}(\mathbf{a})$, for $\mathbf{i}$ satisfying $\mathrm{wt}(\mathbf{i}) = j$. Thus $Q^{(\mathbf{i})}(\mathbf{a}) = 0$ for each $\mathbf{a}$ and each $\mathbf{i}$ with $\mathrm{wt}(\mathbf{i}) < s$.

Therefore $\mathsf{mult}(Q, \mathbf{a}) \geq s$ for each $\mathbf{a} \in \mathbb{F}_q^m$, as desired. □

6 Interpolating sets for multiplicity codes

In this section, we describe an explicit *interpolating set* for multiplicity codes. Concretely, for a given d, s, m, q, we find a set $S \subseteq \mathbb{F}_q^m \times \{\mathbf{i} \mid \mathrm{wt}(\mathbf{i}) < s\}$ such that for every $f : S \to \mathbb{F}_q$, there is *exactly one* $Q(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ of degree at most d such that for each $(\mathbf{a}, \mathbf{i}) \in S$, $Q^{(\mathbf{i})}(\mathbf{a}) = f(\mathbf{a}, \mathbf{i})$. By simple linear-algebra arguments it can be seen that such sets S exist, and any such set S must have $|S| = \binom{d+m}{m}$. Our goal here is to describe an explicit such set; this will enable the local decoding of multiplicity codes in *uniform* sublinear time.

The main fact that we will need is that there are explicit interpolating sets for traditional multivariate polynomial codes. Specifically, for each $d < mq$, there is an explicit set $S_d \subseteq \mathbb{F}_q^m$ such that for every $f : S_d \to \mathbb{F}_q$, there is exactly one $A(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ of degree at most d and individual degree $< q$, such that $A(\mathbf{a}) = f(\mathbf{a})$ for each $\mathbf{a} \in S_d$.

Before solving the problem in full generality, let us consider a simple special case where we can see what is actually going on. Suppose we have $m = 2$ and $s = 2$; thus are dealing with bivariate polynomials of degree $\leq d < 2q$ and we want to find 3 sets of points $S_0, S_X, S_Y \subseteq \mathbb{F}_q^2$ such that if we learn the values of a degree-d polynomial Q on S_0, the values of $\frac{\partial Q}{\partial X}$ on S_X and the values of $\frac{\partial Q}{\partial Y}$ on S_Y, we will be able to uniquely recover Q. Now $Q(X,Y)$ can be written uniquely as

$$Q(X,Y) = A(X,Y) + (X^q - X)B(X,Y) + (Y^q - Y)C(X,Y),$$

where A, B, C all have X and Y degrees $\leq q-1$, and the total degree of A is at most d, and the total degrees of B, C are at most $d - q$. Given this representation of Q, the coordinates of the codeword of the multiplicity code corresponding to Q are easy to compute (using the product rule):

$$\begin{aligned} Q(\mathbf{a}) &= A(\mathbf{a}), \\ \frac{\partial Q}{\partial X}(\mathbf{a}) &= \frac{\partial A}{\partial X}(\mathbf{a}) - B(\mathbf{a}), \\ \frac{\partial Q}{\partial Y}(\mathbf{a}) &= \frac{\partial A}{\partial Y}(\mathbf{a}) - C(\mathbf{a}). \end{aligned}$$

This motivates the following choice for S_0, S_X and S_Y. We pick S_0 to be an interpolating set for polynomials of total degree $\leq d$ and individual degree $\leq q-1$. We pick S_X and S_Y to be interpolating sets for polynomials of degree $\leq d - q$ and individual degree[3] at most $\leq q-1$. Given evaluations of Q on S_0, we

[3] In our special case this latter condition is vacuous.

get evaluations of A on S_0, and thus we can recover A by choice of S_0. Once we have A, the evaluations of $\frac{\partial Q}{\partial X}$ at the points of S_X give us the evaluations of B on S_X, and thus we can recover B. Similarly we can recover C. Putting this all together, we just found A, B and C, and hence we found Q. Thus we found an interpolating set in this special case. The ideas here generalize quite easily to the general case.

For each vector of nonnegative integers $\mathbf{k} = (k_1, \ldots, k_m)$, introduce the polynomial

$$V_{\mathbf{k}}(\mathbf{X}) = \prod_{j=1}^{m} (X_j^q - X_j)^{k_j}.$$

For every $\mathbf{i} = (i_1, \ldots, i_m)$, a simple calculation shows that unless $\mathbf{i} \geq \mathbf{k}$ coordinate-wise, we have that for every $\mathbf{a} \in \mathbb{F}_q^m$:

$$(V_{\mathbf{k}})^{(\mathbf{i})}(\mathbf{a}) = 0. \tag{6.1}$$

Furthermore,

$$(V_{\mathbf{k}})^{(\mathbf{k})}(\mathbf{a}) = (-1)^{\mathrm{wt}(\mathbf{k})}. \tag{6.2}$$

Let $Q(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ be a polynomial of degree $\leq d$. Then Q can be written as

$$Q(\mathbf{X}) = \sum_{\mathbf{k}:\mathrm{wt}(\mathbf{k}) \leq \frac{d}{q}} A_{\mathbf{k}}(\mathbf{X}) \cdot V_{\mathbf{k}}(\mathbf{X}),$$

for some polynomials $A_{\mathbf{k}}(\mathbf{X})$ with individual degrees at most $q-1$ and total degree at most

$$\min\big(m \cdot (q-1), d - \mathrm{wt}(\mathbf{k}) \cdot q\big).$$

Conversely, any collection of such $A_{\mathbf{k}}(\mathbf{X})$ gives a $Q(\mathbf{X})$ of degree $\leq d$.

We now show how the polynomials $A_{\mathbf{k}}(\mathbf{X})$, and hence the polynomial $Q(\mathbf{X})$, can be recovered from certain explicit evaluations of Q and its derivatives.

We have:

$$\begin{aligned} Q^{(\mathbf{i})}(\mathbf{X}) &= \sum_{\mathbf{k}} (A_{\mathbf{k}} \cdot V_{\mathbf{k}})^{(\mathbf{i})}(\mathbf{X}) \\ &= \sum_{\mathbf{k}} \sum_{\mathbf{j}+\mathbf{j}'=\mathbf{i}} (A_{\mathbf{k}})^{(\mathbf{j})}(\mathbf{X}) \cdot (V_{\mathbf{k}})^{(\mathbf{j}')}(\mathbf{X}), \end{aligned}$$

where the last step follows from the product rule for higher-order Hasse derivatives [18, Pages 144-155].

Substituting $\mathbf{a} \in \mathbb{F}_q^m$ into this equation, and using Equation (6.1) and Equation (6.2), we get:

$$Q^{(\mathbf{i})}(\mathbf{a}) = \sum_{\mathbf{k}} \sum_{\mathbf{j}+\mathbf{j}'=\mathbf{i}, \mathbf{j}' \geq \mathbf{k}} (A_{\mathbf{k}})^{(\mathbf{j})}(\mathbf{a}) (V_{\mathbf{k}})^{(\mathbf{j}')}(\mathbf{a}) \tag{6.3}$$

$$= (-1)^{\mathrm{wt}(\mathbf{i})} \cdot A_{\mathbf{i}}(\mathbf{a}) + \sum_{\substack{\mathbf{k} \neq \mathbf{i} \\ \mathbf{k} \leq \mathbf{i}}} \sum_{\substack{\mathbf{j}' \\ \mathbf{i} \geq \mathbf{j}' \geq \mathbf{k}}} (A_{\mathbf{k}})^{(\mathbf{i}-\mathbf{j}')}(\mathbf{a}) (V_{\mathbf{k}})^{(\mathbf{j}')}(\mathbf{a}). \tag{6.4}$$

Thus we can find the evaluation of $A_{\mathbf{i}}(X)$ at the point $\mathbf{a}$ of $\mathbb{F}_q^m$ if we know

- $Q^{(\mathbf{i})}(\mathbf{a})$,
- the polynomial $A_{\mathbf{k}}(X)$ for each $\mathbf{k} < \mathbf{i}$.

This motivates the following construction of an interpolating set for the multiplicity code of order-s evaluations of degree-d polynomials in m variables over $\mathbb{F}_q$.

- For each $\mathbf{i}$ with $\mathrm{wt}(\mathbf{i}) \le d/q$, we choose a set $S_{\mathbf{i}} \subseteq \mathbb{F}_q^m$ which is an interpolating set for the space of all polynomials with individual degree at most $q-1$ and total degree at most $d - \mathrm{wt}(\mathbf{i})q$.
- Then the interpolating set S is defined by:

$$S = \bigcup_{\mathbf{i}:\mathrm{wt}(\mathbf{i})<s} S_{\mathbf{i}} \times \{\mathbf{i}\}\,.$$

We now show that for every $f : S \to \mathbb{F}_q$, there is exactly one $Q(X_1,\dots,X_m) \in \mathbb{F}_q[X_1,\dots,X_m]$ of degree at most d such that $f(\mathbf{a},\mathbf{i}) = Q^{(\mathbf{i})}(\mathbf{a})$ for each $(\mathbf{a},\mathbf{i}) \in S$.

We do this by giving a procedure to find Q given $Q^{(\mathbf{i})}(\mathbf{a})$ for $(\mathbf{a},\mathbf{i}) \in S$.

1. Let $\mathbf{i}$ vary in the set $\{\mathbf{i} \mid \mathrm{wt}(\mathbf{i}) < d/q\}$ in non-decreasing order.
2. By the time we come to a given $\mathbf{i}$, we would have already found $A_{\mathbf{i}'}(\mathbf{X})$ for every $\mathbf{i}' < \mathbf{i}$.
3. Via (6.4), using the known values of $Q^{(\mathbf{i})}(\mathbf{a})$ for $\mathbf{a} \in S_{\mathbf{i}}$ and the known polynomials $A_{\mathbf{i}'}(\mathbf{X})$ ($\mathbf{i}' < \mathbf{i}$), we can find the evaluations of $A_{\mathbf{i}}(\mathbf{a})$ for each $\mathbf{a} \in S_{\mathbf{i}}$.
4. By the interpolation property of $S_{\mathbf{i}}$, this suffices to find $A_{\mathbf{i}}(\mathbf{X})$.

This procedure gives us $A_{\mathbf{i}}(\mathbf{X})$ for each $\mathbf{i}$ with $\mathrm{wt}(\mathbf{i}) \le d/q$, and thus $Q(\mathbf{X})$ is uniquely determined.

7 Local decoding and local list-decoding

In this section, we point out some corollaries to local decoding and local list-decoding of multiplicity codes.

Using the algorithm for unique-decoding of bivariate multiplicity codes of distance δ from a $\delta/2$ fraction of errors, we will get a local-decoding algorithm for decoding m-variate multiplicity codes of distance δ from a $(\delta/2-\varepsilon)$ fraction of errors for every constant $\varepsilon > 0$. Given a received word $r : \mathbb{F}_q^m \to \Sigma$, let us show how to locally correct it at the coordinate $\mathbf{a} \in \mathbb{F}_q^m$. Suppose Q is the polynomial of degree $\le d$ whose corresponding codeword of the multiplicity code is within distance $\delta/2-\varepsilon$ from r.

Local Decoder:

- Let $M = \binom{m+s-1}{m}$.
- Pick lines $\ell_1,\dots,\ell_M$ passing through $\mathbf{a}$ uniformly and independently.
- For each ℓ_i, let $\mathfrak{p}_i$ be a uniformly random 2-dimensional plane containing ℓ_i.

- By querying all the points of $\mathfrak{p}_i$, and decoding from a $\delta/2$ fraction of errors, recover the restriction of Q to the plane $\mathfrak{p}_i$.
- Using this, we get the restriction of Q to each ℓ_i.
- Use this to recover $Q^{(<s)}(\mathbf{a})$.

We briefly sketch the analysis. For any $i \in [M]$, notice that $\mathfrak{p}_i$ is a uniformly random plane passing through $\mathbf{a}$. Thus with high probability, $\mathfrak{p}_i$ will have a $< \delta/2$ fraction of errors on it. By the union bound, this will in fact happen for all the $i \in [M]$ simultaneously with high probability, and thus all the computations of Q restricted to $\mathfrak{p}_i$ and ℓ_i will be correct, and thus so will the computation of $Q^{(<s)}(\mathbf{a})$ (provided the ℓ_i are in general enough position). Making this analysis formal can be done easily following the analogous analysis in [22].

The local list-decoding algorithm involves a little more work. Recall that a local list-decoding algorithm uses oracle access to a received word r and outputs a collection of oracles, such that for each codeword in the list of codewords near r, some oracle in the output locally computes that codeword. We describe the algorithm below:

Local List-decoder:

1. Set $M = \binom{m+s-1}{m}$.
2. Pick lines $\ell_1, \dots, \ell_M$ uniformly at random in $\mathbb{F}_q^m$.
3. For each $i \in [M]$, list-decode the univariate multiplicity code on $r|_{\ell_i}$ from a $(1-\sqrt{1-\delta}-\varepsilon/2)$ fraction of errors to get the list of univariate polynomials $\mathcal{R}_i$.
4. For each $(R_1, \dots, R_M) \in \prod_{i\in[M]} \mathcal{R}_i$, output the oracle $\mathsf{Correct}(A_{(\ell_1,\dots,\ell_M),(R_1,\dots,R_M)})$.

Here Correct is the local correcting algorithm of [22]; when given as input an oracle which computes a received word within distance $0.1\cdot\delta$ from a codeword of a multiplicity code, it simulates an oracle which computes that codeword with high probability.

Next we describe the oracle A. It takes as advice a collection of lines and a collection of univariate polynomials restricted to those lines, and simulates oracle access to a function that is supposed to be $\ll 0.1\cdot\delta$-close to a codeword of the multiplicity code; specifically it is supposed to compute the unique codeword (if any) in the list of codewords close to r whose restriction to line ℓ_i equals the univariate polynomial R_i.

Oracle $A_{(\ell_1,\dots,\ell_M),(R_1,\dots,R_M)}(\mathbf{a})$:

1. For each $i \in [M]$:
   (a) Consider the plane $\mathfrak{p}_i$ containing ℓ_i and $\mathbf{a}$.
   (b) List-decode $r|_{\mathfrak{p}_i}$ from a $(1-\sqrt{1-\delta}-\varepsilon/2)$ fraction of errors to obtain a list $\mathcal{L}_i$ of bivariate polynomials defined on $\mathfrak{p}_i$.
   (c) Choose the unique element (if any) of the list whose restriction to ℓ_i equals R_i; call that bivariate polynomial P_i.

2. Output the unique value of $Q^{(<s)}(\mathbf{a})$ that is consistent with the equations $Q|_{\mathfrak{p}_i} = P_i$.

The analysis of this algorithm is a combination of the local decoding algorithm for multiplicity codes [22] and the local list-decoding algorithms for Reed-Muller codes [1, 27, 2]. The use of planes in the decoding is to allow for decoding from a $(1-\sqrt{1-\delta}+\varepsilon)$ fraction of errors, and not a $(1-\sqrt{2(1-\delta)}-\varepsilon)$ fraction of errors which is what decoding along lines seems to give (this is the extra ingredient in [2] which removes the 2 that appears in [27]; for codes of rate approaching 1, the removal of the 2 is crucial for this algorithm to be nontrivial).

We sketch the analysis now. Let $\hat{Q}(\mathbf{X}) \in \mathbb{F}_q[\mathbf{X}]$ be a polynomial corresponding to a codeword of the multiplicity code. The first step is to show that with high probability, for each $i \in [M]$, $\hat{Q}|_{\ell_i} \in \mathcal{R}_i$. This follows easily from the fact that random lines in $\mathbb{F}_q^m$ "sample well." The second step is to show that with high probability over the choice of $\ell_1, \dots, \ell_M$, the oracle

$$B := A_{(\ell_1,\dots,\ell_M),(\hat{Q}|_{\ell_1},\dots,\hat{Q}|_{\ell_M})}$$

computes a function which is $(\varepsilon^{\Omega(1)})$-close (and hence $\ll (0.1\delta)$-close) to $\hat{Q}$. To see this, we take a random $\mathbf{a} \in \mathbb{F}_q^m$, and show that the probability (over the choice of the ℓ_i) that $B(\mathbf{a}) = \hat{Q}^{(<s)}(\mathbf{a})$ is at least $1-\varepsilon^{\Omega(1)}$. Now the event "$B(\mathbf{a}) = \hat{Q}^{(<s)}(\mathbf{a})$" will occur if: (1) $\hat{Q}|_{\mathfrak{p}_i}$ appears in $\mathcal{L}_i$, (2) no other element of $\mathcal{L}_i$ equals $\hat{Q}|_{\ell_i}$ when restricted to ℓ_i, and (3) the $\mathfrak{p}_i$ are in general enough position. The probability of (1) is $1-\varepsilon^{\Omega(1)}$ because $\mathfrak{p}_i$ is a random plane containing $\mathbf{a}$, and hence it "samples well." The probability of (2) is also $1-\varepsilon^{\Omega(1)}$, because we can think of this event as choosing $\mathfrak{p}_i$ first and then ℓ_i within it, and then its probability is bounded by the probability that some two elements of $\mathcal{L}_i$, which we know is of size $\leq \mathrm{poly}(1/\varepsilon)$, agree on the uniformly random line ℓ_i. This is small by a general bound on the number of lines on which two distinct bivariate polynomials can agree. The probability of (3) is $1-o(1)$. Summing up, we see that the probability of $B(\mathbf{a}) = \hat{Q}^{(<s)}(\mathbf{a})$ is at least $1-\varepsilon^{\Omega(1)}$, and hence we have the desired property of B. Thus the self-corrected oracle $\mathsf{Correct}(B)$ will agree with $\hat{Q}$ on all points of $\mathbb{F}_q^m$, and so the codeword of the multiplicity code corresponding to $\hat{Q}$ will appear in the list of output oracles with high probability, as desired.

8 Open questions

1. Can univariate multiplicity codes be list-decoded from list-decoding capacity with constant list-size? The results of Guruswami, Guruswami-Wang and Dvir-Lovett imply that a carefully chosen (nonlinear) subset of univariate multiplicity codes does have this property.

2. Can bivariate multiplicity codes of minimum distance δ be list-decoded from a $(\delta-\varepsilon)$ fraction of errors with constant list-size? This would automatically translate into a strong local list-decodability for multivariate multiplicity codes. In particular, this would give for every fixed integer m, codes of arbitrarily long length n, rate $(1-\delta)^m$ and distance δ, which are locally list-decodable from a $(\delta-\varepsilon)$ fraction of errors in time $O(n^{2/m})$.

3. Does list-decoding of multiplicity codes have applications in complexity theory? List-decoding traditional polynomial codes (in many variables) has many applications. At first glance, the

advantages of multiplicity codes over polynomial codes seem to be significant only when the number of variables is much smaller.

Acknowledgements

I am very grateful to Professor Enrico Bombieri for generously sharing his insights on a previous version of Theorem 4.3 which worked only for trivariate polynomials, and for suggestions that were instrumental in the proof of the full Theorem 4.3. Thanks to Boaz Barak, Michael Forbes, Amir Shpilka, Madhu Sudan and other members of Madhu's reading group for valuable suggestions during a talk in February, 2011 that further simplified the proof of Theorem 4.3. Thanks also to Shubhangi Saraf, Madhu Sudan and Sergey Yekhanin for helpful discussions and encouragement. Finally, many thanks to the anonymous referees for very helpful comments.

References

[1] SANJEEV ARORA AND MADHU SUDAN: Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003. Preliminary version in STOC'97. See also at ECCC. [doi:10.1007/s00493-003-0025-0] 154, 178

[2] KRISTIAN BRANDER AND SWASTIK KOPPARTY: List-decoding Reed-Muller over large fields upto the Johnson radius. *Manuscript*, 2009. 154, 178

[3] ZEEV DVIR, SWASTIK KOPPARTY, SHUBHANGI SARAF, AND MADHU SUDAN: Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. *SIAM J. Comput.*, 42(6):2305–2328, 2013. Preliminary version in FOCS'09. See also at ECCC. [doi:10.1137/100783704] 156

[4] ZEEV DVIR AND SHACHAR LOVETT: Subspace evasive sets. In *Proc. 44th STOC*, pp. 351–358. ACM Press, 2012. See also at arXiv. [doi:10.1145/2213977.2214010] 155

[5] GERALD B. FOLLAND: *Introduction to Partial Differential Equations*. Princeton University Press, 1995. 152

[6] ALAN GUO: High rate locally correctable codes via lifting. *Electron. Colloq. on Comput. Complexity (ECCC)*, 20:53, 2013. See also at arXiv. 155

[7] ALAN GUO AND SWASTIK KOPPARTY: List-decoding algorithms for lifted codes. *CoRR*, abs/1412.0305, 2014. [arXiv:1412.0305] 155

[8] ALAN GUO, SWASTIK KOPPARTY, AND MADHU SUDAN: New affine-invariant codes from lifting. In ROBERT D. KLEINBERG, editor, *Proc. 4th Conf. on Innovations in Theoretical Computer Science (ITCS'13)*, pp. 529–540. ACM Press, 2013. See also at arXiv and ECCC. [doi:10.1145/2422436.2422494] 155

[9] VENKATESAN GURUSWAMI AND SWASTIK KOPPARTY: Explicit subspace designs. *Combinatorica*, pp. 1–25, 2014. Preliminary version in FOCS'13. See also ECCC. [doi:10.1007/s00493-014-3169-1] 155

[10] VENKATESAN GURUSWAMI AND ATRI RUDRA: Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inform. Theory*, 54(1):135–150, 2008. Preliminary version in STOC'06. [doi:10.1109/TIT.2007.911222] 150, 151, 155

[11] VENKATESAN GURUSWAMI, AMIT SAHAI, AND MADHU SUDAN: "Soft-decision" decoding of Chinese Remainder Codes. In *Proc. 41st FOCS*, pp. 159–168. IEEE Comp. Soc. Press, 2000. [doi:10.1109/SFCS.2000.892076] 153, 168

[12] VENKATESAN GURUSWAMI AND MADHU SUDAN: Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Trans. Inform. Theory*, 45(6):1757–1767, 1999. Preliminary version in FOCS'98. See also at ECCC. [doi:10.1109/18.782097] 151, 153

[13] VENKATESAN GURUSWAMI AND CAROL WANG: Linear-algebraic list decoding for variants of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 59(6):3257–3268, 2013. See also at ECCC. [doi:10.1109/TIT.2013.2246813] 155, 158

[14] VENKATESAN GURUSWAMI AND CHAOPING XING: Folded codes from function field towers and improved optimal rate list decoding. In *Proc. 44th STOC*, pp. 339–350. ACM Press, 2012. See also at ECCC and arXiv. [doi:10.1145/2213977.2214009] 155

[15] VENKATESAN GURUSWAMI AND CHAOPING XING: List decoding Reed-Solomon, Algebraic-Geometric, and Gabidulin subcodes up to the Singleton bound. In *Proc. 45th STOC*, pp. 843–852. ACM Press, 2013. See also at ECCC. [doi:10.1145/2488608.2488715] 155

[16] VENKATESAN GURUSWAMI AND CHAOPING XING: Optimal rate algebraic list decoding using narrow ray class fields. *J. Combin. Theory Ser. A*, 129:160–183, 2015. Partial preliminary version in SODA'14. See also at arXiv and ECCC. [doi:10.1016/j.jcta.2014.09.003] 155

[17] BRETT HEMENWAY, RAFAIL OSTROVSKY, AND MARY WOOTTERS: Local correctability of expander codes. In FEDOR V. FOMIN, RUSINS FREIVALDS, MARTA Z. KWIATKOWSKA, AND DAVID PELEG, editors, *Proc. 40th Internat. Colloq. on Automata, Languages and Programming (ICALP'13)*, volume 7965 of *Lecture Notes in Computer Science*, pp. 540–551. Springer, 2013. See also at arXiv. [doi:10.1007/978-3-642-39206-1_46] 155

[18] JAMES W. P. HIRSCHFELD, GÁBOR KORCHMÁROS, AND FERNANDO TORRES: *Algebraic Curves over a Finite Field (Princeton Series in Applied Mathematics)*. Princeton University Press, 2008. 175

[19] SWASTIK KOPPARTY: *Algebraic methods in randomness and pseudorandomness*. Ph. D. thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2010. DSpace@MIT. 171

[20] SWASTIK KOPPARTY: List-decoding multiplicity codes. *Electron. Colloq. on Comput. Complexity (ECCC)*, 12:44, 2012. 149, 155

[21] SWASTIK KOPPARTY: Some remarks on multiplicity codes. In ALEXANDER BARG AND OLEG R. MUSIN, editors, *Discrete Geometry and Algebraic Combinatorics: AMS Spec. Session*, volume 625 of *Contemporary Mathematics*, pp. 155–176. Amer. Math. Soc., 2014. [doi:10.1090/conm/625, arXiv:1505.07547] 155

[22] SWASTIK KOPPARTY, SHUBHANGI SARAF, AND SERGEY YEKHANIN: High-rate codes with sublinear-time decoding. *J. ACM*, 61(5/28):1–20, 2014. Preliminary version in STOC'11. [doi:10.1145/2629416] 150, 151, 154, 157, 177, 178

[23] FARZAD PARVARESH AND ALEXANDER VARDY: Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proc. 46th FOCS*, pp. 285–294. IEEE Comp. Soc. Press, 2005. [doi:10.1109/SFCS.2005.29] 151

[24] RUUD PELLIKAAN AND XIN-WEN WU: List decoding of q-ary Reed-Muller codes. *IEEE Trans. Inform. Theory*, 50(4):679–682, 2004. [doi:10.1109/TIT.2004.825043] 153

[25] MADHU SUDAN: Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997. Preliminary version in FOCS'96. [doi:10.1006/jcom.1997.0439] 151

[26] MADHU SUDAN: Ideal error-correcting codes: Unifying algebraic and number-theoretic algorithms. In SERDAR BOZTAS AND IGOR SHPARLINSKI, editors, *Proc. 14th Internat. Symp. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC'01)*, volume 2227 of *Lecture Notes in Computer Science*, pp. 36–45. Springer, 2001. [doi:10.1007/3-540-45624-4_4] 153

[27] MADHU SUDAN, LUCA TREVISAN, AND SALIL P. VADHAN: Pseudorandom generators without the XOR lemma. *J. Comput. System Sci.*, 62(2):236–266, 2001. Preliminary version in CCC'99 and STOC'99. See also at ECCC. [doi:10.1006/jcss.2000.1730] 154, 178

[28] SALIL P. VADHAN: Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012. [doi:10.1561/0400000010] 155

AUTHOR

Swastik Kopparty
Department of Mathematics & Department of Computer Science
Rutgers University, Piscataway, NJ
swastik.kopparty@rutgers.edu
`http://www.math.rutgers.edu/~sk1233/`

ABOUT THE AUTHOR

SWASTIK KOPPARTY received his Ph. D. from M.I.T. in 2010, and was advised by Madhu Sudan. He was a postdoc at the Institute for Advanced Study. Earlier, he had been an undergrad at IIT Madras and the University of California, Riverside. At UCR he got interested in theoretical computer science and coding theory, largely due to the guidance of Ilya Dumer and Chinya Ravishankar. Swastik's fields of research include $\mathbb{F}_2$, $\mathbb{F}_p$, $\mathbb{F}_q$, $\mathbb{R}$ and $\mathbb{C}$. He is also interested in complexity theory, error-correcting codes, table tennis, randomness and pseudorandomness.