Weight Distribution and List-Decoding Size of Reed–Muller Codes

Tali Kaufman, Shachar Lovett, and Ely Porat

***Abstract*—The weight distribution and list-decoding size of Reed–Muller codes are studied in this work. Given a weight parameter, we are interested in bounding the number of Reed–Muller codewords with weight up to the given parameter; and given a received word and a distance parameter, we are interested in bounding the size of the list of Reed–Muller codewords that are within that distance from the received word. Obtaining tight bounds for the weight distribution of Reed–Muller codes has been a long standing open problem in coding theory, dating back to 1976. In this work, we make a new connection between computer science techniques used to study low-degree polynomials and these coding theory questions. This allows us to resolve the weight distribution and list-decoding size of Reed–Muller codes for all distances. Previous results could only handle bounded distances: Azumi, Kasami, and Tokura gave bounds on the weight distribution which hold up to 2.5 times the minimal distance of the code; and Gopalan, Klivans, and Zuckerman gave bounds on the list-decoding size which hold up to the Johnson bound.**

***Index Terms*—List decoding, Reed–Muller codes, weight distributions.**

I. Introduction

THE weight distribution of an error correcting code counts, for every given weight parameter, the number of codewords with weight bounded by the given parameter. The weight distribution of a code is the main characteristic of the code and governs the behavior of the code, from both theoretical and practical aspects.

Understanding the weight distribution of Reed–Muller codes is a 30-year-old open problem in coding theory. The last progress on this question was made by Kasami and Tokura [10] who characterized the codewords of Reed–Muller codes of weight up to twice the minimal distance of the code, and hence obtained bounds for the weight distribution that apply up to twice the minimal distance of the code; and by Azumi, Kasami, and Tokura [11] who obtained bounds on codewords of weight up to 2.5 times the minimal distance. In this work we resolve this problem and obtain asymptotically tight bounds for the weight distribution that apply to all distances.

The problem of list-decoding an error correcting code is the following: given a received word and a distance parameter find all codewords that are within the given distance from the received word. List-decoding is a generalization of the more common notion of unique decoding in which the given distance parameter ensures that there can be at most one codeword that is within the given distance from the received word. List-decoding has many applications for both practical and theoretical problems. We refer the interested reader to surveys by Guruswami [7] and Sudan [15] on list decoding. See [3]–[5], [8], [13], [14] for works dealing with list-decoding of Reed–Muller codes.

In this paper, we study the question of list-decoding Reed–Muller codes. Specifically, we are interested in bounding the list size for any distance parameter. Previous results of Gopalan, Klivans, and Zuckerman [6] gave bounds on the list-decoding size for distance parameters which are not too large (specifically, distances up to the Johnson bound). In this work we obtain asymptotically tight bounds for the list-decoding size that apply to all distances.

Our results are obtained by making a new connection between computer science techniques used for studying low-degree polynomials and the discussed coding theoretic problems. This connection allows to us to analyze the weight distribution and list-decoding size in a relatively simple way. We view this as evidence for the importance of this connection.

A. Reed–Muller Codes: Facts and Previous Bounds

Reed–Muller codes form a basic and well studied family of codes. $\mathrm{RM}(n,d)$ is a linear code, whose codewords $f \in \mathrm{RM}(n,d) : \mathbb{F}_2^n \to \mathbb{F}_2$ are evaluations of polynomials in n variables of total degree at most d over $\mathbb{F}_2$. In this work we study the code $\mathrm{RM}(n,d)$ when $d \ll n$, and are interested in particular in the case of constant d.

The following facts regarding $\mathrm{RM}(n,d)$ are straightforward: It has block length of 2^n, dimension $\sum_{i \le d} \binom{n}{i}$ and minimum relative distance $\frac{2^{n-d}}{2^n} = 2^{-d}$. We next discuss the weight distribution and list-decoding size of Reed–Muller codes.

Weight Distribution of Reed–Muller Codes: The relative (or normalized) weight of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the relative number of ones in it

$$\mathrm{wt}(f) := \frac{1}{2^n} |\{x \in \mathbb{F}_2^n : f(x) = 1\}|.$$

Manuscript received April 13, 2010; revised September 15, 2011; accepted September 20, 2011. Date of publication January 31, 2012; date of current version April 17, 2012. This work was supported in part by the Alon Fellowship and by the National Science Foundation under Grant DMS-0835373. The work of T. Kaufman was supported in part by NSF Awards CCF-0514167 and NSF-0729011. The work of S. Lovett was supported in part by the Israel Science Foundation (Grant 1300/05). This work was performed in part when the S. Lovett was an intern at Microsoft Research.

T. Kaufman is with Bar-Ilan University, Ramat Gan, Israel. He is also with The Weizmann Institute of Science, Rehovot 76100 Israel (e-mail: kaufmant@mit.edu).

S. Lovett is with the Institute for Advanced Study, Faculty of Mathematics and Computer Science, and The Weizmann Institute of Science, Rehovot 76100, Israel (e-mail: slovett@math.ias.edu; shachar.lovett@gmail.com).

E. Porat is with the Department of Computer Science, Bar-Ilan University, Ramat Gan, Israel (e-mail: porately@cs.biu.ac.il).

Communicated by J.-P. Tillich, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2012.2184841

The cumulative weight distribution of $\mathrm{RM}(n,d)$ at a relative weight $0 \leq \alpha \leq 1$, denoted $W_{n,d}(\alpha)$, is the number of codewords whose relative weight is at most α,

$$W_{n,d}(\alpha) := |\{f \in \mathrm{RM}(n,d) : \mathrm{wt}(f) \leq \alpha\}|.$$

The fact that the minimal relative distance of $\mathrm{RM}(n,d)$ is 2^{-d} implies that $W_{n,d}(2^{-d}-\varepsilon) = 1$ for any $\varepsilon > 0$. Kasami and Tokura [10] characterized the codewords in $\mathrm{RM}(n,d)$ of weight up to twice the minimal distance of the code (i.e., up to relative distance 2^{1-d}). Based on this characterization Gopalan *et al.* deduce the following upper bound.

Corollary I.1 [6, Cor. 10] $W_{n,d}(2^{1-d}-\varepsilon) \leq (1/\varepsilon)^{2(n+1)}$*:*

Combining this upper bound with a simple lower bound, one gets that $W_{n,d}(2^{1-d}-\varepsilon)$ is exponential in n for any constant $0 < \varepsilon \leq 2^{-d}$. On the other extreme, as half of the codewords in $\mathrm{RM}(n,d)$ have relative weight at most $1/2$, we have that $W_{n,d}(1/2)$ is exponential in n^d. Nontrivial upper bounds on the cumulative weight distribution for weight parameters $2^{1-d} \leq \alpha < 1/2$ were unknown prior to this work.

List-Decoding Size of Reed–Muller Codes: The relative distance between two functions $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ is the fraction of inputs on which they differ

$$\mathrm{dist}(f,g) = \frac{1}{2^n}|\{x \in \mathbb{F}_2^n : f(x) \neq g(x)\}|.$$

The *list-decoding size* of $\mathrm{RM}(n,d)$ at a relative distance $0 \leq \alpha \leq 1$, denoted $L_{n,d}(\alpha)$, is the maximal number of codewords within relative distance α from any word

$$L_{n,d}(\alpha) = \max_{g:\mathbb{F}_2^n \to \mathbb{F}_2} |\{f \in \mathrm{RM}(n,d) : \mathrm{dist}(f,g) \leq \alpha\}|.$$

It is clear that the list-decoding size is lower bounded by the cumulative weight distribution. Gopalan *et al.* [6] obtain bounds on the list decoding size for bounded distance parameters. They show that the list size is constant for distances up to the minimal distance, and polynomial (in the block length 2^n) for distances up to a quantity which is bounded by twice the minimal distance.

Theorem I.2 [6, Th. 11]:

1. $L_{n,d}(2^{-d}-\varepsilon) \leq O\left((1/\varepsilon)^{8d}\right)$.
2. $L_{n,d}(J(2^{1-d})-\varepsilon) \leq (1/2\varepsilon^2)^{n+O(1)}$ where $J(\alpha) := \frac{1}{2}(1-\sqrt{1-2\alpha}) \leq \alpha$ is the Johnson bound.

Moreover, the work of Gopalan *et al.* develops a general list-decoding algorithm for $\mathrm{RM}(n,d)$ whose running time is polynomial in the list-decoding size. Thus, they reduce the algorithmic list-decoding problem to the combinatorial problem of bounding the list size.

B. Our Results

We give tight bounds on the cumulative weight distribution and list-decoding size for Reed–Muller codes for relative weights and distances beyond the minimal distance. We show that the cumulative weight distribution and the list-decoding size have similar asymptotic behavior: they are both exponential in n^ℓ for integer values of ℓ which depend on the parameter α. Moreover, the value of ℓ jumps at the same "cut-off" relative weights or distances. In the following the asymptotic notation $O_d(\cdot)$, $\Omega_d(\cdot)$ hides constants which depend only on d.

Theorem 3.1 (Main Result, Informal Statement): Let $2^{-d} \leq \alpha < 1/2$ be a parameter. Let $1 \leq k \leq d-1$ and $0 < \varepsilon \leq 1/2$ be such that $\alpha = 2^{-k}(1-\varepsilon)$. Then

$$(1/\varepsilon)^{\Omega_d(n^{d-k})} \leq W_{n,d}(\alpha) \leq L_{n,d}(\alpha) \leq (1/\varepsilon)^{O_d(n^{d-k})}.$$

We observe that the bound given by Theorem 3.1 is tight even for sub-constant values of ε: the minimal value of ε is 2^{-n}, at which stage the bound becomes exponential in n^{d-k+1}.

C. Techniques

Our main result, Theorem 3.1, combines matching lower and upper bounds. The lower bound follows from a simple construction. Let $\alpha = 2^{-k}(1-\varepsilon)$ for $\varepsilon = 2^{-e}$. We assume w.l.o.g that $e \leq n/10$ as otherwise the bound will follow from the bound for $\alpha = 2^{-k}(1-2^{-n/10})$. Let $p_1, \ldots, p_e$ be arbitrary polynomials of degree $d-k$ in the variables $x_{k+2e+1}, \ldots, x_n$ and set

$$f(x) := \prod_{i=1}^{k-1} x_i \cdot \Big(\sum_{j=1}^{e} x_{k+j} \cdot (p_j(x) + x_{k+e+j})\Big).$$

It is not hard to verify that f has relative weight $2^{-k}(1-\varepsilon)$. The number of different choices for f is dictated by the number of choices for $p_1, \ldots, p_e$ which is $2^{\Omega_d(n^{d-k})\cdot e} = (1/\varepsilon)^{\Omega_d(n^{d-k})}$.

Our main contribution is the upper bound given in Theorem 3.1. The main technical ingredient used in the proof is the use of directional derivatives. For $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and a direction $y \in \mathbb{F}_2^n$ the directional derivative of f in direction y is $\Delta_y f(x) := f(x+y) - f(x)$. Iterated direction derivative in directions $Y = \{y_1, \ldots, y_k\}$ is given by $\Delta_Y f = \Delta_{y_1} \ldots \Delta_{y_k} f$. We call $\Delta_Y f$ for $|Y| = k$ a k-iterated derivative of f. Note that if f is a degree d polynomial, then its derivatives have degree at most $d-1$ and its k-iterated derivatives have degree at most $d-k$.

Our main technical lemma shows that any function of small relative weight can be approximated by an algorithm which has oracle access to a small number if its iterated derivatives.

Lemma 2.1 (Informal Statement): Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\varepsilon)$ for $0 < \varepsilon < 1$ and let $\delta > 0$ be an approximation parameter. There exists a universal algorithm $\mathcal{A}$ (which does not depend of f) with the following properties:

1. $\mathcal{A}$ has two inputs: $x \in \mathbb{F}_2^n$ and t sets $Y_1, \ldots, Y_t$ of k directions each.
2. $\mathcal{A}$ has oracle access to the k-iterated derivatives $\Delta_{Y_1} f(\cdot), \ldots, \Delta_{Y_t} f(\cdot)$.

Then for $t = O(\log(1/\delta)\log(1/\varepsilon) + \log(1/\delta)^2)$ there exists a setting for $Y_1, \ldots, Y_t$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[\mathcal{A}(x; Y_1, \ldots, Y_t, \Delta_{Y_1} f(\cdot), \ldots, \Delta_{Y_t} f(\cdot)) = f(x)] \geq 1-\delta.$$

We first describe how the upper bound in Theorem 3.1 follows from Lemma 2.1. Consider for simplicity a bound on the number of codewords in $\mathrm{RM}(n,d)$ with Hamming weight at most $2^{-k}(1-\varepsilon)$. Let δ be small enough to be determined later, and consider the family of all possible functions computed by $\mathcal{A}$ applied to some codeword $f \in \mathrm{RM}(n,d)$ (i.e., some degree d polynomial). Let $\mathcal{H}$ denote this family of functions. We have the following two properties:

1. If $\mathrm{wt}(f) \leq 2^{-k}(1-\varepsilon)$ then there exists $h \in \mathcal{H}$ which $1-\delta$ approximates f.
2. If we choose δ small enough ($\delta < 2^{-d-1}$ suffices), then the minimal distance of $\mathrm{RM}(n,d)$ guarantees that each $h \in \mathcal{H}$ can $1-\delta$ approximate at most one codeword $f \in \mathrm{RM}(n,d)$.

The combination of these properties guarantees that $W_{n,d}(2^{-k}(1-\varepsilon)) \leq |\mathcal{H}|$. Now, the size of $\mathcal{H}$ is dominated by the number of possibilities for $\Delta_{Y_1}f(\cdot),\ldots,\Delta_{Y_t}f(\cdot)$. As each k-iterated derivative is a polynomial of degree $d-k$ the number of choices for it is exponential in n^{d-k}; and as $t = O_d(\log(1/\varepsilon))$ we get that $|\mathcal{H}| \leq 2^{O_d(n^{d-k})\cdot\log(1/\varepsilon)} = (1/\varepsilon)^{O_d(n^{d-k})}$ as claimed.

The bound for the list-decoding size follows a similar approach. Let $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be some fixed function where we want to bound the number of codewords in $\mathrm{RM}(n,d)$ which have distance at most $2^{-k}(1-\varepsilon)$ from g. Define $\mathcal{H}$ as the family of functions computed by $\mathcal{A}$ applied to $f \oplus g$ where f ranges over all codewords in $\mathrm{RM}(n,d)$ and apply the same argument as before.

We now describe the proof of Lemma 2.1. The goal is to show that a function with low weight can be approximated by an algorithm with access to a small number of its derivatives. More concretely, if $\mathrm{wt}(f) = 2^{-k}(1-\varepsilon)$ we wish to show that f can be approximated by an algorithm with access to $O_d(\log(1/\varepsilon))$ many k-derivatives of f.

Consider first the case of $k=1$, i.e., of a function f with $\mathrm{wt}(f) = \frac{1}{2}(1-\varepsilon)$. That is, f is an ε-biased function. A lemma of Bogdanov and Viola [2] shows any ε-biased function can be approximated by a simple function of $(1/\varepsilon)^{O(1)}$ of its derivatives. To get the correct dependency on ε, we effectively derandomize their result and show that a similar function with access to only $O(\log(1/\varepsilon))$ derivatives achieves the same result. For $k>1$ we follow a two step process: we first show that f can be approximated by a function of a constant number of its $k-1$ derivatives; and that each $k-1$ derivative is by itself ε-biased. We then follow the approach for $k=1$ to further approximate each $k-1$ derivative by $O(\log(1/\varepsilon))$ of its derivatives, which are k-derivatives of f.

Organization: The paper is organized as follows. In Section II we prove the main technical lemma, showing that a low-weight function can be approximated by its iterated derivatives. We then apply this lemma to bounding the weight distribution and list-decoding size of Reed–Muller codes in Section III. We study the extension of our techniques for Generalized Reed–Muller codes in Section IV, where we provide some (non tight) bounds for these codes.

II. Approximation of Low-Weight Functions by Derivatives

We prove in this section that any low-weight function can be approximated by a function with access to a small number of its derivatives. We first recall some definitions from the introduction. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function. The relative (or normalized) *weight* of f is the fraction of ones in f,

$$\mathrm{wt}(f) = \Pr_{x\in\mathbb{F}_2^n}[f(x)=1].$$

The *bias* of f is given by

$$\mathrm{bias}(f) = \mathbb{E}_{x\in\mathbb{F}_2^n}[(-1)^{f(x)}] = 1-2\mathrm{wt}(f).$$

The derivative of f in direction $y \in \mathbb{F}_2^n$ is $\Delta_y f(x) := f(x+y)+f(x)$; and the k-iterated derivative of f in directions $Y = (y_1,\ldots,y_k) \in (\mathbb{F}_2^n)^k$ is

$$\Delta_Y f(x) := \Delta_{y_1}\ldots\Delta_{y_k}f(x) = \sum_{I\subseteq[k]} f(x+\sum_{i\in I} y_i).$$

Note that the order of $y_1,\ldots,y_k$ is irrelevant for the definition of $\Delta_Y f$, and so we can think of Y as a multi-set of size k.

The following lemma is the main result proved in this section.

Lemma 2.1: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\varepsilon)$ for $0<\varepsilon<1$ and let $\delta>0$ be an approximation parameter. There exists a universal algorithm $\mathcal{A}$ (which does not depend of f) with the following properties:

1. $\mathcal{A}$ has two inputs: $x \in \mathbb{F}_2^n$ and $Y_1,\ldots,Y_t \in (\mathbb{F}_2^n)^k$.
2. $\mathcal{A}$ has oracle access to the k-iterated derivatives $\Delta_{Y_1}f(\cdot),\ldots,\Delta_{Y_t}f(\cdot)$.

Then for $t = O(\log(1/\delta)\log(1/\varepsilon)+\log(1/\delta)^2)$ there exists a setting for $Y_1,\ldots,Y_t$ such that

$$\Pr_{x\in\mathbb{F}_2^n}[\mathcal{A}(x;Y_1,\ldots,Y_t,\Delta_{Y_1}f(\cdot),\ldots,\Delta_{Y_t}f(\cdot)) = f(x)] \geq 1-\delta.$$

The proof follows two main steps. First, for $k \geq 2$ we show that f can be approximated by a majority function of $O(\log(1/\delta))$ many $(k-1)$-iterated derivatives of f. Crucially, the number of derivatives used does not depend on ε.

Lemma 2.2: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}$ for $k \geq 2$. Then there exist sets of directions $Y_1,\ldots,Y_a \in (\mathbb{F}_2^n)^{k-1}$ where $a = O(\log(1/\delta))$ such that

$$\Pr[f(x) = \mathrm{Maj}(\Delta_{Y_1}f(x),\ldots,\Delta_{Y_a}f(x))] \geq 1-\delta.$$

We prove Lemma 2.2 in Section 2.1. We next note that each $(k-1)$-iterated derivative of f is biased.

Claim 2.3: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\varepsilon)$. Then for any $Y \in (\mathbb{F}_2^n)^{k-1}$ we have $\mathrm{bias}(\Delta_Y f) \geq \varepsilon > 0$.

Proof: Let $Y = (y_1,\ldots,y_{k-1})$ and recall that $\Delta_Y f(x) = \sum_{I\subseteq[k-1]} f(x+\sum_{i\in I} y_i)$. That is, $\Delta_Y f$ is a sum of 2^{k-1} shifted versions of f. Hence $\mathrm{wt}(\Delta_Y f) \leq 2^{k-1}\mathrm{wt}(f) \leq \frac{1}{2}(1-\varepsilon)$ which gives $\mathrm{bias}(\Delta_Y f) = 1-2\mathrm{wt}(\Delta_Y f) \geq \varepsilon$. □

We thus proceed to approximate biased functions by an algorithm with query access to a small number of their derivatives. Bogdanov and Viola [2] show that any ε-biased function can be approximated by a function of $(1/\varepsilon)^{O(1)}$ of its derivatives. Here we refine their argument and show that such a function can in fact be approximated by an algorithm with oracle access to $O(\log(1/\varepsilon))$ of the derivatives.

Lemma 2.4: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{bias}(f) \geq \varepsilon > 0$ and let $\delta > 0$ be an approximation parameter. There exists a universal algorithm $\mathcal{A}'$ (which does not depend of f) with the following properties:

1. $\mathcal{A}'$ has two inputs: $x \in \mathbb{F}_2^n$ and $y_1,\ldots,y_b \in \mathbb{F}_2^n$.

2. $\mathcal{A}'$ has oracle access to the derivatives $\Delta_{y_1} f(\cdot), \ldots, \Delta_{y_b} f(\cdot)$.

Then for $b = O(\log(1/\varepsilon) + \log(1/\delta))$ there exists a setting for $y_1, \ldots, y_b$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[\mathcal{A}'(x; y_1, \ldots, y_b, \Delta_{y_1} f(\cdot), \ldots, \Delta_{y_b} f(\cdot)) = f(x)] \geq 1 - \delta.$$

We prove Lemma 2.4 in Section II-B. The proof of Lemma 2.1 now follows immediately from a combination of Lemma 2.2, Claim 2.3, and Lemma 2.4.

Proof of Lemma 2.1: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1 - \varepsilon)$. Apply Lemma 2.2 with error parameter δ^2. There exist $a = O(\log(1/\delta))$ sets of directions $Y_1, \ldots, Y_a \in (\mathbb{F}_2^n)^{k-1}$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[f(x) = \mathrm{Maj}(\Delta_{Y_1} f(x), \ldots, \Delta_{Y_a} f(x))] \geq 1 - \delta^2.$$

Moreover, by Claim 2.3 we know that $\mathrm{bias}(\Delta_{Y_i} f) \geq \varepsilon$ for all $1 \leq i \leq a$. Thus, we can apply Lemma 2.4 to each $\Delta_{Y_i} f$ independently with error parameter δ^2, getting that there exists a setting of $y_{i,1}, \ldots, y_{i,b} \in \mathbb{F}_2^n$ with $b = O(\log(1/\varepsilon\delta))$ such that (see the equation at the bottom of the page). Let $Y_{i,j} \in (\mathbb{F}_2^n)^k$ be formed by concatenating Y_i and $y_{i,j}$. Define the algorithm $\mathcal{A}$ to simulate this two-step process. That is, the algorithm $\mathcal{A}$ has inputs $x \in \mathbb{F}_2^n$ and $Y_{i,j} \in (\mathbb{F}_2^n)^k$ and oracle access to $\Delta_{Y_{i,j}} f(\cdot)$. The algorithm $\mathcal{A}$ ranges over $i \in [a]$; it simulates $\mathcal{A}'(x; y_1, \ldots, y_{i,b}, \Delta_{y_{i,1}} \Delta_{Y_i} f(\cdot), \ldots, \Delta_{y_{i,b}} \Delta_{Y_i} f(\cdot)))$ using the values of $y_{i,1}, \ldots, y_{i,b}$ and oracle access to $\Delta_{Y_{i,j}} f(\cdot)$; and then combines the answers using majority. By the union bound we have that

$$\begin{aligned}\Pr_{x \in \mathbb{F}_2^n}[f(x) &= \mathcal{A}(x; \{Y_{i,j}\}, \{\Delta_{Y_{i,j}} f(\cdot)\})] \\ &\geq 1 - \delta^2 \cdot (a+1) = 1 - \delta^2 \cdot O(\log(1/\delta)) \\ &\geq 1 - \delta\end{aligned}$$

for δ bounded by some absolute constant. □

A. Proof of Lemma 2.2

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function with $\mathrm{wt}(f) \leq 2^{-k}$ for $k \geq 2$ and let $\delta > 0$ be an error parameter. We wish to show that there exist $a = O(\log(1/\delta))$ sets of directions $Y_1, \ldots, Y_a \in (\mathbb{F}_2^n)^{k-1}$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[f(x) = \mathrm{Maj}(\Delta_{Y_1} f(x), \ldots, \Delta_{Y_a} f(x))] \geq 1 - \delta.$$

We first show that $f(x)$ can be computed as a weighted average of its derivatives with bounded coefficients.

Lemma 2.5: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}$ for $k \geq 2$. Then there exist coefficients $\{\alpha_Y : Y \in (\mathbb{F}_2^n)^{k-1}\}$ where $0 \leq \alpha_Y \leq 10$ such that

$$(-1)^{f(x)} = \mathbb{E}_{Y \in (\mathbb{F}_2^n)^{k-1}}[\alpha_Y \cdot (-1)^{\Delta_Y f(x)}].$$

The proof of Lemma 2.5 is based on an iterative application of the following claim from [12] which is a simplification of a lemma in [2]. It shows that a biased function can be computed by the average of its derivatives.

Claim 2.6: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{bias}(f) \neq 0$. Then

$$(-1)^{f(x)} = \frac{1}{\mathrm{bias}(f)} \mathbb{E}_{y \in \mathbb{F}_2^n}[(-1)^{\Delta_y f(x)}].$$

For completeness, we give the proof.

Proof: Fix x and let $y \in \mathbb{F}_2^n$ be chosen uniformly. We have

$$\begin{aligned}\mathbb{E}_{y \in \mathbb{F}_2^n}[(-1)^{\Delta_y f(x)}] &= \mathbb{E}_{y \in \mathbb{F}_2^n}[(-1)^{f(x+y)+f(x)}] \\ &= (-1)^{f(x)} \mathbb{E}_{y \in \mathbb{F}_2^n}[(-1)^{f(x+y)}] \\ &= (-1)^{f(x)} \cdot \mathrm{bias}(f).\end{aligned}$$

□

We now prove Lemma 2.5.

Proof of Lemma 2.5: Applying Claim 2.6 iteratively $k-1$ times to $f, \Delta_{y_1} f, \ldots, \Delta_{y_1, \ldots, y_{k-2}} f$ we get that

$$(-1)^{f(x)} = \mathbb{E}_{y_1, \ldots, y_{k-1} \in \mathbb{F}_2^n}[\alpha_{y_1, \ldots, y_{k-1}} (-1)^{\Delta_{y_1, \ldots, y_{k-1}} f(x)}],$$

where $\alpha_{y_1, \ldots, y_{k-1}} = \frac{1}{\mathrm{bias}(f)} \cdot \frac{1}{\mathrm{bias}(\Delta_{y_1} f)} \cdot \ldots \cdot \frac{1}{\mathrm{bias}(\Delta_{y_1, \ldots, y_{k-2}} f)}$. To conclude we need to show that $|\alpha_{y_1, \ldots, y_{k-1}}| \leq 10$. Note that for any $\ell \leq k-2$, $\Delta_{y_1, \ldots, y_\ell} f = \sum_{I \subseteq [\ell]} f(x + \sum_{i \in I} y_i)$ hence $\mathrm{wt}(\Delta_{y_1, \ldots, y_\ell} f) \leq 2^\ell \cdot \mathrm{wt}(f) \leq 2^{\ell - k}$. Thus

$$\begin{aligned}\frac{1}{\alpha_{y_1, \ldots, y_{k-1}}} &= \mathrm{bias}(f) \cdot \mathrm{bias}(\Delta_{y_1} f) \cdot \ldots \cdot \mathrm{bias}(\Delta_{y_1, \ldots, y_{k-2}} f) \\ &= (1 - 2\mathrm{wt}(f)) \cdot (1 - 2\mathrm{wt}(\Delta_{y_1} f)) \cdot \ldots \\ &\quad \cdot (1 - 2\mathrm{wt}(\Delta_{y_1, \ldots, y_{k-2}} f)) \\ &\geq \prod_{\ell=0}^{k-2} (1 - 2^{\ell - k + 1}) \\ &\geq \prod_{m=1}^{\infty} (1 - 2^{-m}) \geq 1/10.\end{aligned}$$

□

The proof of Lemma 2.2 follows from Lemma 2.5 by a standard sampling argument.

$$\Pr_{x \in \mathbb{F}_2^n}[\Delta_{Y_i} f(x) = \mathcal{A}'(x; y_{i,1}, \ldots, y_{i,b}, \Delta_{y_{i,1}} \Delta_{Y_i} f(\cdot), \ldots, \Delta_{y_{i,b}} \Delta_{Y_i} f(\cdot))] \geq 1 - \delta^2.$$

Proof of Lemma 2.2: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \le 2^{-k}$. Applying Lemma 2.5 we have

$$(-1)^{f(x)} = \mathbb{E}_Y[\alpha_Y \cdot (-1)^{\Delta_Y f(x)}]$$

where Y is uniformly chosen in $(\mathbb{F}_2^n)^{k-1}$ and where $0 \le \alpha_Y \le 10$. Let $\Omega = (\mathbb{F}_2^n)^{k-1}$ denote the set of possible Y. Let $c = |\Omega|^{-1} \sum_{Y \in \Omega} \alpha_Y$. By assumption $c \le 10$. Define a probability distribution $\mathcal{D}$ over Ω by choosing Y with probability proportional to α_Y, i.e., $\Pr_{\mathcal{D}}[Y] = \frac{\alpha_Y}{c|\Omega|}$. Note that under this distribution we have

$$(-1)^{f(x)} = c \cdot \mathbb{E}_{Y \sim D}[(-1)^{\Delta_Y f(x)}].$$

Thus, we must have $c \ge 1$ and

$$(-1)^{f(x)} \cdot c^{-1} = \mathbb{E}_{Y \sim D}[(-1)^{\Delta_Y f(x)}].$$

Let $Y_1, \ldots, Y_a$ be sampled according to $\mathcal{D}$ independently. Note that if

$$\left| \frac{c \sum_{i=1}^a (-1)^{\Delta_{Y_i} f(x)}}{a} - (-1)^{f(x)} \right| < 1$$

then

$$\begin{aligned}(-1)^{f(x)} &= \mathrm{sign}\left(\frac{c \sum_{i=1}^a (-1)^{\Delta_{Y_i} f(x)}}{a} \right) \\ &= \mathrm{sign}\left(\sum_{i=1}^a (-1)^{\Delta_{Y_i} f(x)} \right) \\ &= (-1)^{\mathrm{Maj}(\Delta_{Y_1} f(x), \ldots, \Delta_{Y_a} f(x))}.\end{aligned}$$

We next apply standard Chernoff bounds (see, e.g., $A.1.16$ in [1]) and get that by setting $a = O(c^2 \log(1/\delta)) = O(\log(1/\delta))$ we have for any fixed x that

$$\Pr_{Y_1, \ldots, Y_a \sim D}[f(x) = \mathrm{Maj}(\Delta_{Y_1} f(x), \ldots, \Delta_{Y_a} f(x))] \ge 1 - \delta.$$

Thus, by an averaging argument there exists a setting for $Y_1, \ldots, Y_a$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[f(x) = \mathrm{Maj}(\Delta_{Y_1} f(x), \ldots, \Delta_{Y_a} f(x))] \ge 1 - \delta.$$

Indeed, for showing the above it suffices to show that

$$\Pr_{Y_1, \ldots, Y_a \sim D}\left[\left| \frac{c \sum_{i=1}^a (-1)^{\Delta_{Y_i} f(x)}}{a} - (-1)^{f(x)} \right| > \frac{1}{4} \right] < \delta.$$

Consider random variables $z_1, \ldots, z_a$ defined as

$$z_i = \frac{c}{c+1}\left[(-1)^{\Delta_{Y_i} f(x)} - \frac{(-1)^{f(x)}}{c} \right].$$

We have $\mathbb{E}[z_i] = 0$; $|z_i| \le 1$; and $z_1, \ldots, z_a$ are mutually independent. Thus, by $A.1.16$ in [1],

$$\Pr_{z_1, \ldots, z_a \sim D}\left[\sum_{i=1}^a z_i > r \right] < e^{\frac{-r^2}{2a}}.$$

By setting $r = \frac{a}{4(c+1)}$ we get that

$$\begin{aligned}\Pr_{Y_1, \ldots, Y_a \sim D}&\left[\left| \frac{c \sum_{i=1}^a (-1)^{\Delta_{Y_i} f(x)}}{a} - (-1)^{f(x)} \right| > \frac{1}{4} \right] \\ &= \Pr_{z_1, \ldots, z_a \sim D}\left[\sum_{i=1}^a z_i > r \right] < e^{\frac{-a}{32(c+1)^2}}.\end{aligned}$$

Thus, by setting $a = O(c^2 \log(1/\delta)) = O(\log(1/\delta))$ we get the claimed bound. □

B. Proof of Lemma 2.4

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{bias}(f) \ge \varepsilon > 0$ and let $\delta > 0$ be an error parameter. We wish to show that there exists a universal algorithm $\mathcal{A}'$ (which does not depend on f) that gets as inputs $x \in \mathbb{F}_2^n$, a set of directions $y_1, \ldots, y_b \in \mathbb{F}_2^n$ and has oracle access to the derivatives of f in these directions, such that for some setting of $y_1, \ldots, y_b$ we have

$$\Pr_{x \in \mathbb{F}_2^n}[\mathcal{A}'(x; y_1, \ldots, y_b, \Delta_{y_1} f(\cdot), \ldots, \Delta_{y_b} f(\cdot)) = f(x)] \ge 1 - \delta.$$

The algorithm is simple. For $I \subseteq [b]$ let $y_I := \sum_{i \in I} y_i$ denote the partial sums of $y_1, \ldots, y_b$. The algorithm computes the majority of $\Delta_{y_I} f$, where I ranges over for all nonempty subsets of $[b]$

$$\begin{aligned}&\mathcal{A}'(x; y_1, \ldots, y_b, \Delta_{y_1} f(\cdot), \ldots, \Delta_{y_b} f(\cdot)) \\ &\qquad = \mathrm{Maj}(\{\Delta_{y_I} f(x) : \emptyset \ne I \subseteq [b]\}).\end{aligned}$$

To prove the correctness of the algorithm we need to prove two things: first, that the algorithm can in fact compute the majority of $\{\Delta_{y_I} f(x)\}$ given its inputs; and second, that this majority indeed approximates f well for some setting of the directions $y_1, \ldots, y_b$. We first show that the algorithm can indeed compute the required majority.

Claim 2.7: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function and let $y_1, \ldots, y_b \in \mathbb{F}_2^n$ a set of directions. Then for any $x \in \mathbb{F}_2^n$ and any nonempty $I \subseteq [b]$, there is an algorithm which computes the value of $\Delta_{y_I} f(x)$ given as inputs $x, I, y_1, \ldots, y_b$ and oracle access to $\Delta_{y_1} f(\cdot), \ldots, \Delta_{y_b} f(\cdot)$.

Proof: Let $I = \{i_1, \ldots, i_r\}$. The algorithm computes the value of $\Delta_{y_I} f(x)$ using the following identity:

$$\Delta_{y_I} f(x) = \sum_{\ell=1}^r \Delta_{y_{i_\ell}} f(x + \sum_{j=1}^{\ell-1} y_{i_j}).$$

□

We next show that $f(x)$ can be approximated by the majority of $\Delta_{y_I} f(x)$ where I ranges over non-empty subsets of $[b]$ for some choice of $y_1, \ldots, y_b$. In fact, we show that most choices of $y_1, \ldots, y_b$ are suitable.

Claim 2.8: Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{bias}(f) \ge \varepsilon > 0$ and let $\delta > 0$ be an error parameter. Then

$$\begin{aligned}\Pr_{x, y_1, \ldots, y_b \in \mathbb{F}_2^n}&[f(x) = \mathrm{Maj}(\{\Delta_{y_I} f(x) : \emptyset \ne I \subseteq [b]\})] \\ &\ge 1 - \delta\end{aligned}$$

for $b = O(\log(1/\varepsilon) + \log(1/\delta))$. In particular, there is a setting for $y_1, \ldots, y_b$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[f(x) = \mathrm{Maj}(\{\Delta_{y_I} f(x) : \emptyset \neq I \subseteq [b]\})] \geq 1 - \delta.$$

Proof: Define

$$S(x, y_1, \ldots, y_b) := \sum_{\emptyset \neq I \subseteq [b]} (-1)^{f(x) + \Delta_{y_I} f(x)} = \sum_{\emptyset \neq I \subseteq [b]} (-1)^{f(x + y_I)}$$

and note that $S > 0$ iff $f(x) = \mathrm{Maj}(\{\Delta_{y_I} f(x) : \emptyset \neq I \subseteq [b]\}$. Thus, we need to show that $\Pr_{x, y_1, \ldots, y_b \in \mathbb{F}_2^n}[S > 0] \geq 1 - \delta$ for a suitable choice of b. We will use Chebychev's inequality. The expected value of S is

$$\mathbb{E}[S] = (2^b - 1)\mathbb{E}[(-1)^f] = (2^b - 1)\mathrm{bias}(f)$$

and as the summands in S are pairwise independent (for a uniform choice of $x, y_1, \ldots, y_b \in \mathbb{F}_2^n$) the variance of S is bounded by

$$\mathrm{Var}[S] = (2^b - 1)\mathrm{Var}[(-1)^f] \leq (2^b - 1)\mathrm{bias}(f).$$

We thus conclude that

$$\Pr[S \leq 0] \leq \Pr[|S - \mathbb{E}[S]| \geq \mathbb{E}[S]] \leq \frac{\mathrm{Var}[S]}{|\mathbb{E}[S]|^2} \leq \frac{1}{(2^b - 1)\mathrm{bias}(f)}.$$

Since $\mathrm{bias}(f) \geq \varepsilon$, for $b = O(\log(1/\varepsilon) + \log(1/\delta))$ we conclude that $\Pr[S \leq 0] \leq \delta$. □

III. Bounds for Reed–Muller Codes

In this section, we bound the cumulative weight distribution and list-decoding size of Reed–Muller codes. The following is our main result.

Theorem 3.1 (Main Result): Let $2^{-d} \leq \alpha < 1/2$ be a parameter. Let $1 \leq k \leq d - 1$ and $0 < \varepsilon \leq 1/2$ be such that $\alpha = 2^{-k}(1 - \varepsilon)$. Then

$$(1/\varepsilon)^{c_d \cdot n^{d-k}} \leq W_{n,d}(\alpha) \leq L_{n,d}(\alpha) \leq (1/\varepsilon)^{C_d \cdot n^{d-k}}$$

where $c_d, C_d > 0$ are constants which depend only on d.

We prove the lower and upper bound in the Sections III-B and III-C.

A. Proof of Lower Bound

We prove a lower bound on the cumulative weight distribution $W_{n,d}(\alpha)$. The lower bound follows from a simple construction. Let $\alpha = 2^{-k}(1 - \varepsilon)$ for $\varepsilon = 2^{-e}$. We assume w.l.o.g that $e \leq n/10$ as otherwise the bound will follow from the bound for $\alpha = 2^{-k}(1 - 2^{-n/10})$.

Let $p_1, \ldots, p_e$ be arbitrary polynomials of degree $d - k$ in the variables $x_{k+2e+1}, \ldots, x_n$ and consider polynomials of the form

$$f(x_1, \ldots, x_n) := \prod_{i=1}^{k-1} x_i \cdot (\sum_{j=1}^{e} x_{k+j} \cdot (p_j(x) + x_{k+e+j})).$$

We clearly have $f \in \mathrm{RM}(n, d)$. We would shortly show that $\mathrm{wt}(f) = 2^{-k}(1 - \varepsilon)$. The number of distinct such f is a lower bound on $W_{n,d}(\alpha)$. This is the number of choices for each polynomial $p_i(x)$ (which is exponential in $\binom{n-(k+2e)+(d-k)}{d-k} = \frac{1}{(d-k)!}\Omega(n)^{d-k}$) raised to the power e. That is

$$W_{n,d}(\alpha) \geq 2^{\frac{1}{(d-k)!}O(n^{d-k}) \cdot \log(1/\varepsilon)} = (1/\varepsilon)^{\frac{1}{(d-k)!}O(n^{d-k})}.$$

The computation of the relative weight of f follows immediately from the following claim and the fact the the polynomials $p_1, \ldots, p_e$ do not contain the variables $x_1, \ldots, x_{k+2e}$.

Claim 3.2: Let $a_1, \ldots, a_e \in \mathbb{F}_2$ be constants. Let $q(x_1, \ldots, x_{k+2e})$ be the polynomial

$$q(x_1, \ldots, x_{k+2e}) := \prod_{i=1}^{k-1} x_i \cdot (\sum_{j=1}^{e} x_{k+j} \cdot (a_j + x_{k+e+j})).$$

Then $\Pr[q = 1] = 2^{-k}(1 - 2^{-e})$.

Proof: Let $y_j = x_{k+j} \cdot (a_j + x_{k+e+j})$. Note that $\Pr[y_j = 1] = 1/4$, that $x_1, \ldots, x_{k-1}, y_1, \ldots, y_e$ are independent and that $q(x) := \prod_{i=1}^{k-1} x_i \cdot (\sum_{j=1}^{e} y_j)$. Thus

$$\Pr[q = 1] = \prod_{i=1}^{k-1} \Pr[x_i = 1] \cdot \Pr[\sum_{j=1}^{e} y_j = 1] = 2^{1-k} \cdot \frac{1}{2}(1 - \mathbb{E}[(-1)^{y_1 + \cdots + y_e}]) = 2^{-k}(1 - \prod_{j=1}^{e} \mathbb{E}[(-1)^{y_j}]) = 2^{-k}(1 - 2^{-e}).$$

□

B. Proof of Upper Bound

We prove an upper bound on the list-decoding size $L_{n,d}(\alpha)$. Let $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be fixed. We wish to bound the number of codewords $f \in \mathrm{RM}(n, d)$ for which $\mathrm{dist}(f, g) \leq \alpha$.

Let $\mathcal{A}$ be the algorithm guaranteed by Lemma 2.1. We would show that any function $f \in \mathrm{RM}(n, d)$ whose distance from g is at most $\alpha = 2^{-k}(1 - \varepsilon)$ can be well approximated by the function computed by $\mathcal{A}$ applied to $f \oplus g$, and then xoring this function to g. Choosing the error parameter δ to be below half the minimal distance of $\mathrm{RM}(n, d)$ would guarantee that each function can approximate at most one codeword. The upper bound follows from an upper bound on the different number of functions that can be computed by the algorithm (recall the g is fixed, hence, is not counted).

Formally, let $\mathcal{H}$ denote the following family of functions. Let $f \in \mathrm{RM}(n,d)$ and $Y_1,\ldots,Y_t \in (\mathbb{F}_2^n)^k$. Let $h(x):\mathbb{F}_2^n \to F_2$ denote the function computed by the algorithm $\mathcal{A}$ applied to $f+g$ given directions $Y_1,\ldots,Y_t$ and error parameter δ to be determined later

$$h(x)=\mathcal{A}(x;Y_1,\ldots,Y_t,\Delta_{Y_1}(f+g)(\cdot),\ldots,\Delta_{Y_i}(f+g)(\cdot)).$$

We define $\mathcal{H}$ to be the family of all such functions h. The family $\mathcal{H}$ has the following two properties:

1. If $\mathrm{dist}(f,g) \le 2^{-k}(1-\varepsilon)$ then there exists $h \in \mathcal{H}$ for which $\Pr_{x\in\mathbb{F}_2^n}[f(x)+g(x)=h(x)] \ge 1-\delta$. This follows from the guarantees of Lemma 2.1 since $\mathrm{wt}(f+g) \le 2^{-k}(1-\varepsilon)$.
2. If we choose δ smaller than half the minimal distance of the code $\mathrm{RM}(n,d)$ (i.e., $\delta < 2^{-d-1}$), then for any function h there can be at most one codeword $f \in \mathrm{RM}(n,d)$ for which $\Pr_{x\in\mathbb{F}_2^n}[f(x)+g(x)=h(x)] \ge 1-\delta$ (if there were two such codewords f_1, f_2 then this would imply that $\mathrm{dist}(f_1,f_2)<\delta$).

Thus the number of functions in $\mathcal{H}$ is an upper bound on $L_{n,d}(\alpha)$. The number of different possibilities for each Y_i is 2^{nk}. Given Y_i, the number of possible functions $\Delta_{Y_i}(f+g)(\cdot)$ is bounded by the number of n-variate polynomials of degree $d-k$, since f is an arbitrary degree d polynomial, g is fixed and by linearity $\Delta_{Y_i}(f+g)(\cdot)=\Delta_{Y_i}(f)(\cdot)+\Delta_{Y_i}(g)(\cdot)$. The number of such polynomials is exponential in n^{d-k}. Thus we get that

$$|\mathcal{H}| \le 2^{(nk+\frac{1}{(d-k)!}n^{d-k})t}.$$

The choice of δ gives $t=O(d^2+d\log(1/\varepsilon))$ which implies

$$|\mathcal{H}| \le (1/\varepsilon)^{O_d(n^{d-k})}.$$

IV. Generalized Reed–Muller Codes

The problems of bounding both the cumulative weight distribution and the list-decoding size can be extended to Generalized Reed–Muller, the code of low-degree polynomials over larger fields. However, our techniques fail to prove tight result in these cases. Following we give some partial results for cumulative weight distribution and the list-decoding size of Generalized Reed–Muller codes.

We start by making some basic definitions. Let q be a prime and let $\mathrm{GRM}_q(n,d)$ denote the code of multivariate polynomials $f(x_1,\ldots,x_n)$ over the field $\mathbb{F}_q$ of total degree at most d.

Definition 4.1: The relative weight of a function $f:\mathbb{F}_q^n \to \mathbb{F}_q$ is the fraction of non-zero elements,

$$\mathrm{wt}(f)=\frac{1}{q^n}|\{x\in\mathbb{F}_q^n : f(x)\neq 0\}|.$$

Definition 4.2: The relative distance between two functions $f,g:\mathbb{F}_q^n \to \mathbb{F}_q$ is defined as

$$\mathrm{dist}(f,g)=\frac{1}{q^n}|\{x\in\mathbb{F}_q^n : f(x)\neq g(x)\}|.$$

The cumulative weight distribution and the list-decoding size are defined analogously for $\mathrm{GRM}_q(n,d)$, using the appropriate definitions for relative weight and relative distance. We denote them by $W_{n,d,q}$ and $L_{n,d,q}$. For each $1 \le k \le d$, we define a relative distance r_k as follows.

1. For $k=1$, let $a \ge 0$, $1 \le b \le q-1$ be such that $d=(q-1)a+b$. Define $r_1=q^{-a}(1-b/q)$. r_1 is the minimal relative distance of $\mathrm{GRM}_q(n,d)$.
2. For $2 \le k \le d-1$, let $a \ge 0$, $1 \le b \le q-1$ be such that $d-k=(q-1)a+b$. Define $r_k=q^{-a}(1-b/q)(1-1/q)$.
3. For $k=d$, define $r_d=1-1/q$.

We conjecture that both for the cumulative weight distribution and the list-decoding size, the distances r_k are the thresholds for the exponential dependency in n.

Conjecture 4.3: Let $\varepsilon > 0$ be constant, and consider $\mathrm{GRM}_q(n,d)$ for constant d. Then

- For $\alpha \le r_1-\varepsilon$ both $W_{n,d,q}(\alpha)$ and $L_{n,d,q}(\alpha)$ are constants. This case was already conjectured by [6].
- For $r_k \le \alpha \le r_{k+1}-\varepsilon$ both $W_{n,d,q}(\alpha)$ and $L_{n,d,q}(\alpha)$ are $2^{\Omega(n^k)}$.
- For $\alpha \ge r_d$ both $W_{n,d,q}(\alpha)$ and $L_{n,d,q}(\alpha)$ are $2^{\Omega(n^d)}$.

Here and in the reminder of this section, asymptotic notation (e.g., $\Omega(\cdot)$) hides constants which depend only on d, q.

Proving lower bounds for $W_{n,d,q}(r_k)$ is similar to the case of $\mathrm{RM}(n,d)$.

Lemma 4.4 (Lower Bound for the Weight Distribution of Generalized Reed–Muller Code): For any integer $1 \le k \le d$

$$W_{n,d,q}(r_k) \ge 2^{\Omega(n^k)}.$$

Proof: We start by proving for $2 \le k \le d-1$. Let $d-k=(q-1)a+b$, where $1 \le b \le q-1$. Single out $a+2$ variables $x_1,\ldots,x_{a+2}$, and let f be any degree k polynomial on the remaining variables. The following polynomial has degree d and weight exactly $q^{-a}(1-b/q)(1-1/q)$:

$$g(x_1,\ldots,x_n) := \Big(\prod_{i=1}^{a}\prod_{j=1}^{q-1}(x_i-j)\Big)\cdot\Big(\prod_{j=1}^{b}(x_{a+1}-j)\Big)\cdot(x_{a+2}+f(x_{a+3},\ldots,x_n)).$$

The number of distinct polynomials f is $2^{\Omega(n^k)}$. The proofs for $k=1$ and $k=d$ are similar. For $k=1$, let $d=(q-1)a+b$. Let $\ell_1(x),\ldots,\ell_{a+1}(x)$ be any independent linear functions, and consider

$$g(x_1,\ldots,x_n) := \Big(\prod_{i=1}^{a}\prod_{j=1}^{q-1}(\ell_i(x)-j)\Big)\Big(\prod_{j=1}^{b}(\ell_{a+1}(x)-j)\Big).$$

For $k=d$, let f be any degree d polynomial on variables $x_2,\ldots,x_n$, and consider $g(x_1,\ldots,x_n) := x_1+f(x_2,\ldots,x_n)$. □

Using directly the derivatives method we used to give upper bounds for $\mathrm{RM}(n,d)$ gives the same bounds for $\mathrm{GRM}_q(n,d)$, alas they are not tight for $q>2$.

$$W_{n,d,q}(2^{-k}-\varepsilon) \le 2^{O(n^{d-k})}.$$

Following, we give partial results for Conjecture 4.3 at both ends of the spectrum. We give results when $\alpha \leq r_1 - \varepsilon$, and when $r_{d-1} \leq \alpha \leq r_d - \varepsilon$ (when $\alpha \geq r_d$ Lemma 4.4 gives $L_{n,d,q}(\alpha)$ and $W_{n,d,q}(\alpha)$ are both exponential in n^d, and this is obviously tight).

First, the minimal distance of $\mathrm{GRM}_q(n,d)$ is known to be r_1. Thus, for any $\varepsilon > 0$, $W_{n,d,q}(r_1 - \varepsilon) = 1$. Gopalan, Klivans and Zuckerman [6] prove that $L_{n,d,q}(r_1 - \varepsilon)$ is constant when $q-1$ divides d.

Theorem 4.5 [6, Corollary 18]: Assume $q-1$ divides d. Then

$$L_{n,d,q}(r_1 - \varepsilon) \leq c(q,d,\varepsilon).$$

Moving to the case of $r_{d-1} \leq \alpha \leq r_d - \varepsilon$, we prove the following.

Lemma 4.6 (Upper Bound for the Weight Distribution of Generalized Reed–Muller Code): Let $\varepsilon > 0$ be constant. then

$$W_{n,d,q}(r_d - \varepsilon) \leq 2^{O(n^{d-1})}.$$

We first make some necessary definitions.

Definition 4.7: The bias of a polynomial $f(x_1, \ldots, x_n)$ over $\mathbb{F}_q$ is defined to be

$$\mathrm{bias}(f) = \mathbb{E}_{x \in \mathbb{F}_q^n}[\omega^{f(x)}]$$

where $\omega = e^{2\pi i/q}$ is a primitive qth root of unity.

Kaufman and Lovett [9] prove that biased low-degree polynomials can be decomposed into a function of a constant number of lower degree polynomials.

Theorem 4.8 [9, Th. 2]: Let $f(x_1, \ldots, x_n)$ be a degree d polynomial, such that $|\mathrm{bias}(f)| \geq \varepsilon$. Then f can be decomposed as a function of a constant number of lower degree polynomials

$$f(x) = F(g_1(x), \ldots, g_c(x))$$

where $deg(g_i) \leq d-1$ and $c = c(q,d,\varepsilon)$.

We will use Theorem 4.8 to bound $W_{n,d,q}(r_d - \varepsilon)$ for any constant $\varepsilon > 0$.

Proof of Lemma 4.6: We will show that any polynomial $f \in \mathrm{GRM}_q(n,d)$ such that $\mathrm{wt}(f) \leq 1 - 1/q - \varepsilon$ can be decomposed as

$$f(x) = F(g_1(x), \ldots, g_c(x))$$

where $deg(g_i) \leq d-1$, and c depends only on q, d and ε. Thus the number of such polynomials is bounded by the number of possibilities to choose c degree $d-1$ polynomials, and a function $F : \mathbb{F}_q^c \to \mathbb{F}_q$. The number of such possibilities is at most $2^{O(n^{d-1})}$. Let f be such that $\mathrm{wt}(f) \leq 1 - 1/q - \varepsilon$. We will show there exists $\alpha \in \mathbb{F}_q$, $\alpha \neq 0$ such that $\mathrm{bias}(\alpha f) \geq \varepsilon$. We will then finish by using Theorem 4.8 on the polynomial αf. Consider the bias of αf for random $\alpha \in \mathbb{F}_q$,

$$\mathbb{E}_{\alpha \in \mathbb{F}_q}[\mathrm{bias}(\alpha f)] = \mathbb{E}_{\alpha \in \mathbb{F}_q, x \in \mathbb{F}_q^n}[\omega^{\alpha f(x)}] = 1 - \mathrm{wt}(f),$$

since for x's for which $f(x) = 0$, $\mathbb{E}_{\alpha \in \mathbb{F}_q}[\omega^{\alpha f(x)}] = 1$, and for x such that $f(x) \neq 0$, $\mathbb{E}_{\alpha \in \mathbb{F}_q}[\omega^{\alpha f(x)}] = 0$. We thus get that

$$\mathbb{E}_{\alpha \in \mathbb{F}_q \setminus \{0\}}[\mathrm{bias}(\alpha f)] = 1 - \frac{q}{q-1}\mathrm{wt}(f) \geq \frac{q}{q-1}\varepsilon.$$

So, there must exist $\alpha \neq 0$ such that $\mathrm{bias}(\alpha f) \geq \varepsilon$ and the lemma follows. □

ACKNOWLEDGMENT

The authors would like to thank M. Sudan for helpful comments on this work. The second author (S. Lovett) would like to thank his advisor, O. Reingold, for on-going advice and encouragement. S. Lovett would like to thank Microsoft Research for their support during his internship.

REFERENCES

[1] A. Noga and H. S. Joel, *The Probabilistic Method*. New York: Wiley, 1992.
[2] A. Bogdanov and E. Viola, "Pseudorandom bits for polynomials," *SIAM J. Comput.*, no. 39, pp. 2464–2486, Apr. 2010.
[3] I. Dumer, G. A. Kabatiansky, and C. Tavernier, "List decoding of biorthogonal codes and the Hadamard transform with linear complexity," *IEEE Tran. Inf. Theory*, vol. 54, no. 10, pp. 4488–4492, 2008.
[4] I. Dumer and K. Shabunov, "Soft-decision decoding of Reed–Muller codes: Recursive lists," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1260–1266, 2006.
[5] R. Fourquet and C. Tavernier, "An improved list decoding algorithm for the second order Reed–Muller codes and its applications," *Des. Codes Cryptogr.*., vol. 49, no. 1–3, pp. 323–340, 2008.
[6] P. Gopalan, A. R. Klivans, and D. Zuckerman, "List-decoding Reed–Muller codes over small fields," in *Proc. 40th Annu. ACM Symp. Theory of Computing*, New York, 2008, pp. 265–274, STOC '08.
[7] V. Guruswami, "List Decoding of Error-Correcting Codes," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 2001.
[8] G. G. Kabatiansky, I. Dumer, and C. Tavernier, "List decoding of Reed–Muller codes up to the johnson bound with almost linear complexity," in *IEEE Int. Symp Information Theory*, Jul. 2006, pp. 138–142.
[9] T. Kaufman and S. Lovett, "Worst case to average case reductions for polynomials," in *49th Annu. IEEE Symp. Foundations of Computer Science, FOCS '08*, Oct. 2008, pp. 166–175.
[10] T. Kasami and N. Tokura, "On the weight structure of Reed–Muller codes," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 6, pp. 752–759, Nov. 1970.
[11] T. Kasami, N. Tokura, and S. Azumi, "On the weight enumeration of weights less than 2.5d of Reed–Muller codes," *Inf. Contr.*, vol. 30, no. 4, pp. 380–395, 1976.
[12] S. Lovett, "Unconditional pseudorandom generators for low degree polynomials," *Theory Comput.*, vol. 5, no. 1, pp. 69–82, 2009.
[13] R. Pellikaan and X.-W. Wu, "List decoding of q-ary Reed–Muller codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 4, pp. 679–682, 2004.
[14] N. Santhi, "On algebraic decoding of q-ary Reed–Muller and Product–Reed–Solomon codes," in *Proc. ISIT 2007*, Jun. 2007, pp. 1351–1355.
[15] M. Sudan, "List decoding: Algorithms and applications," *SIGACT News*, no. 31, pp. 16–27, Mar. 2000.