Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 111 (2008)

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

Tali Kaufman *
The Weizmann Institute of Science
kaufmant@mit.edu

Shachar Lovett †
The Weizmann Institute of Science
shachar.lovett@weizmann.ac.il

Ely Porat
Bar-Ilan University
porately@cs.biu.ac.il

May 28, 2010

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 a weight of up to the given parameter. Additionally, 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. In this work, we make a new connection between computer science techniques used for studying low-degree polynomials and these coding theory questions. Using this connection we progress significantly towards resolving both the weight distribution and the list-decoding problems.

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 and seemingly resistent to the common coding theory tools. The best results to date are by Azumi, Kasami and Tokura which provide bounds on the weight distribution that apply only up to 2.5 times the minimal distance of the code. We provide asymptotically tight bounds for the weight distribution of the Reed-Muller code that apply to *all* distances.

List-decoding has both theoretical and practical applications in various fields. To name a few, hardness amplification in complexity, constructing hard-core predicates from one way functions in cryptography and learning parities with noise in learning theory.

Many algorithms for list-decoding have the crux of their analysis lying in bounding the list-decoding size. The case for Reed–Muller codes is similar, and Gopalan, Klivans and Zuckerman gave a list-decoding algorithm, whose complexity is determined by the list-decoding size. Gopalan et. al provided bounds on the list-decoding size of Reed–Muller codes which apply only up to the minimal distance of the code. We provide asymptotically tight bounds for the list-decoding size of Reed–Muller codes which apply to *all* distances.

*Research supported in part by a Koshland Fellowship.
†Research supported by the Israel Science Foundation (grant 1300/05) and by an ERC grant of Irit Dinur.

ISSN 1433-8092

1 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 standing open question in coding theory. The last progress on this question was made by Kasami and Tokura [11] that 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 till twice the minimal distance of the code. In this work we study the weight distribution of Reed Muller codes and provide asymptotically tight bounds 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 of the code 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 of the code that is within the given distance from the received word. The notion of list-decoding has numerous practical and theoretical implications. The breakthrough results in this field are due to Goldreich and Levin [4] and Sudan [13] who gave efficient list decoding algorithms for the Hadamard code and the Reed-Solomon code. See surveys by Guruswami [8] and Sudan [14] for further details. In complexity, list-decodable codes are used to perform hardness amplification of functions [15]. In cryptography, list-decodable codes are used to construct hard-core predicates from one way functions [4]. In learning theory, list decoding of Hadamard codes implies learning parities with noise [10].

In this work we study the question of list-decoding Reed-Muller codes. Specifically, we are interested in bounding the list sizes obtained for different distance parameters for the list-decoding problem. Our work provides asymptotically tight bounds that apply to all distances. The improved bounds, imply improved algorithms for list-decoding Reed-Muller codes.

Our results are obtained by making a new connection between computer science techniques used for studying low-degree polynomials and the discussed coding theory questions. Using this connection we manage to progress significantly towards resolving these two important open problems.

Our proofs are technically relatively simple. We view this as evidence to the importance of this new connection, since these were considered as open problems, resistent to the more common coding theory tools. We view this as the main innovation of our work.

1.1 Reed–Muller codes

Reed-Muller codes are a very fundamental 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 straight-forward: It has block length of 2^n, dimension $\sum_{i \le d} \binom{n}{i}$ and minimum relative distance $\frac{2^{n-d}}{2^n} = 2^{-d}$.

1.2 Weight distribution of Reed-Muller codes

We now formally define the weight distribution of a code, and discuss previous known bounds for the weight distribution of Reed-Muller codes.

Definition 1 (Relative weight)**.** The relative weight of a function/codeword $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the fraction of non-zero elements,

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

Definition 2 (Accumulative weight distribution)**.** The accumulative weight distribution of $\mathrm{RM}(n, d)$ at a relative weight α is the number of codewords up to this weight, i.e.

$$A(\alpha) = |\{p \in \mathrm{RM}(n, d) : \mathrm{wt}(p) \leq \alpha\}|$$

where $0 \leq \alpha \leq 1$.

It is well-known that for any $p \in \mathrm{RM}(n, d)$ which is not identically zero, $\mathrm{wt}(p) \geq 2^{-d}$. Thus, $A(2^{-d} - \epsilon) = 1$ for any $\epsilon > 0$. Kasami and Tokura [11] characterized the codewords in $\mathrm{RM}(n, d)$ of weight up to twice the minimal distance of the code (i.e up to distance 2^{1-d}). Based on their characterization one could conclude the following.

Corollary 1 (Corollary 10 in [7])**.**

$$A(2^{1-d} - \epsilon) \leq (1/\epsilon)^{2(n+1)}$$

Corollary 1 and simple lower bounds (which we show later, see Lemma 15) show that $A(\alpha) = 2^{\Theta(n)}$ for $\alpha \in [2^{-d}, 2^{1-d} - \epsilon]$ for any $\epsilon > 0$ (and constant d).

1.3 List-decoding size of Reed-Muller codes

We now formally define the list-decoding size of a code, and discuss previous known bounds for the list-decoding size of Reed-Muller codes. Moreover we discuss known list-decoding algorithms for Reed-Muller codes. We start with the following definition.

Definition 3 (Relative distance between two functions)**.** The relative distance between two functions $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as

$$\mathrm{dist}(f, g) = \mathbb{P}_{x \in \mathbb{F}_2^n}[f(x) \neq g(x)]$$

Our work focuses on understanding the asymptotic growth of the list size in list-decoding of Reed-Muller codes, as a function of the distance parameter. Specifically we are interested in obtaining bounds on the following.

Definition 4 (List-decoding size)**.** For a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ let the ball at relative distance α around f be

$$B(f, \alpha) = \{p \in \mathrm{RM}(n, d) : \mathrm{dist}(p, f) \leq \alpha\}$$

The list-decoding size of $\mathrm{RM}(n, d)$ at distance α, denoted by $L(\alpha)$, is the maximal size of $B(f, \alpha)$ over all possible functions f, i.e.

$$L(\alpha) = \max_{f:\mathbb{F}_2^n \to \mathbb{F}_2} |B(f, \alpha)|$$

In a recent work, Gopalan, Klivans and Zuckerman [7] proved that for distances up to the minimal distance of the code, the list-decoding size of Reed-Muller codes remains constant.

Theorem 2 (Theorem 11 in [7])**.**

$$L(2^{-d} - \epsilon) \leq O\left((1/\epsilon)^{8d}\right)$$

Their result of bounding the list-decoding size of Reed-Muller codes is inherently limited to work up to the minimum distance of the code, since it uses the structural theorem of Kasami and Takura on Reed-Muller codes [11], which implies a bound on the weight distribution of Reed-Muller codes that works up to twice the minimum distance of the code.

Additionally, the work of [7] has developed a list-decoding algorithm for $\mathrm{RM}(n, d)$ whose running time is polynomial in the worst list-decoding size and in the block length of the code.

Theorem 3 (Theorem 4 in [7])**.** *Given a distance parameter α and a received word $R : \mathbb{F}_2^n \to \mathbb{F}_2$, there is an algorithm that runs in time poly$(2^n, L(\alpha))$ and produces a list of all $p \in \mathrm{RM}(n, d)$ such that* $\mathrm{dist}(p, R) \leq \alpha$.

Since Gopalan et al. could obtain non-trivial bounds on the list-decoding size for distance parameter α that is bounded by the minimum distance of the Reed-Muller code, their algorithm running time could be analyzed only for α that is less than the minimum distance of the code. This supports our earlier statement, that the crux of the analysis of list-decoding algorithms is in bounding the list-decoding size.

1.4 Our Results

The weight distribution of $\mathrm{RM}(n, d)$ codes beyond twice the minimum distance was widely open prior to our work. See e.g. Research Problem (15.1) in [12] and the related discussion in that chapter. In this work we provide asymptotic bounds for the weight distribution of $\mathrm{RM}(n, d)$ that applied for all weights $2^{-d} \leq \alpha \leq 1/2$. We state now our results for constant d, where the notation $O(\cdot), \Omega(\cdot), \Theta(\cdot)$ hides constants depending only on d. Our first main result gives exact boundaries on the range of α for which $A(\alpha) = 2^{\Theta(n^\ell)}$, for any $\ell = 1, 2, ..., d$, showing there are "cut-off distances", at which the accumulative weight distribution jumps from $2^{\Theta(n^\ell)}$ to $2^{\Theta(n^{\ell+1})}$.

Theorem 4 (First main theorem - accumulative weight distribution)**.** *Let $1 \leq \ell \leq d-1$ be an integer, and let $\epsilon > 0$. For any $\alpha \in [2^{\ell-d-1}, 2^{\ell-d} - \epsilon]$*

$$2^{\Omega(n^\ell)} \leq A(\alpha) \leq (1/\epsilon)^{O(n^\ell)}$$

and $A(\alpha) = 2^{\Theta(n^d)}$ for any $\alpha \geq 1/2$.

We also address the more general problem of bounding the list-decoding size. Gopalan et al. [7] left as an open problem the question of bounding the list-decoding size of Reed-Muller codes beyond the minimal distance. We give tight bounds on the list-decoding size of Reed–Muller codes that apply to all distances. In fact, we show that the behavior of the list-decoding size is asymptotically identical to that of the accumulative weight distribution.

Theorem 5 (Second main theorem - list-decoding size)**.** *Let $1 \leq \ell \leq d-1$ be an integer, and let $\epsilon > 0$. For any $\alpha \in [2^{\ell-d-1}, 2^{\ell-d} - \epsilon]$*

$$2^{\Omega(n^\ell)} \leq L(\alpha) \leq (1/\epsilon)^{O(n^\ell)}$$

and $L(\alpha) = 2^{\Theta(n^d)}$ for any $\alpha \geq 1/2$.

Using Theorem 5 and Theorem 3, we obtain the following algorithmic result for list-decoding Reed-Muller codes.

Theorem 6 (List-decoding algorithm)**.** *Let $R : \mathbb{F}_2^n \to \mathbb{F}_2$ be a received word. Let $\alpha \in [2^{\ell-d-1}, 2^{\ell-d} - \epsilon]$ be a required distance parameter, where $1 \leq \ell \leq d-1$ is integer and $\epsilon > 0$. There exists an algorithm that runs in time $(1/\epsilon)^{O(n^\ell)}$ and produces a list of all $p \in \mathrm{RM}(n,d)$ such that $\mathrm{dist}(p,R) \leq \alpha$.*

Observe that Theorems 4 and 5 are asymptotically tight even for sub-constant values of ϵ. The smallest possible value is $\epsilon = 2^{-n}$, and indeed for $\alpha = 2^{\ell-d} - \epsilon$ we get that both $A(\alpha)$ and $L(\alpha)$ are upper bounded by $(1/\epsilon)^{O(n^\ell)} = 2^{O(n^{\ell+1})}$, while for $\alpha = 2^{\ell-d}$ they are lower bounded by $2^{O(n^{\ell+1})}$.

1.5 Techniques

Our results are obtained by making a new connection between computer science techniques used for studying low-degree polynomials and weight distribution and list-decoding size of Reed-Muller codes. Evidence of the importance of this new connection is the technical simplicity of our proofs that solve these well-known open problems. Following is a detailed discussion of our techniques.

The bounds on the accumulative weight distribution of the Reed-Muller code are obtained using the following novel strategy. We study the structure of functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ based on their discrete derivatives. The discrete derivative of f in direction $y \in \mathbb{F}_2^n$ is given by

$$f_y(x) = f(x+y) + f(x).$$

The k-iterated derivative of f in directions $y_1, \ldots, y_k \in \mathbb{F}_2^n$ is given by

$$f_{y_1,\ldots,y_k}(x) = (f_{y_1,\ldots,y_{k-1}})_{y_k}(x) = \sum_{I \subseteq \{1,\ldots,k\}} f(x + \sum_{i \in I} y_i).$$

Note that if f is an n-variate polynomial over $\mathbb{F}_2$ of total degree d, then any derivative of it is a polynomial of total degree at most $d-1$, and any k-iterated derivative of it is a polynomial of total degree at most $d-k$. This is an important property that is crucial to our proof.

As a first step to bounding the weight distribution of Reed-Muller codes we establish the following general result. We show that a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ whose weight is bounded by $\mathrm{wt}(f) \leq 2^{-k}(1-\epsilon)$ can be approximated by a universal function $\mathcal{A}$ of a small number of the k-iterated derivatives of f (Lemma 7). That is, for any approximation parameter $\delta > 0$, there exist $c = c(k, \epsilon, \delta)$ sets of k-iterated derivatives $\{y_{i,1}, \ldots, y_{i,k}\}_{1 \leq i \leq c}$, such that

$$\Pr_{x \in \mathbb{F}_2^n}[f(x) \neq \mathcal{A}(\{y_{i,j}\}, f_{y_{1,1},\ldots,y_{1,k}}(x), \ldots, f_{y_{c,1},\ldots,y_{c,k}}(x))] < \delta.$$

We accomplish this in three steps. It will be useful to represent functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ as $(-1)^f : \mathbb{F}_2^n \to \{-1,1\}$. First, we show that the function $(-1)^{f(x)}$ can be *computed* as an expectation of its $(k-1)$-iterated derivatives $(-1)^{f_{y_1,\ldots,y_{k-1}}(x)}$ multiplied by some bounded coefficients (Lemma 8). Moreover, we show that each of the $(k-1)$-iterated derivatives is biased (a function $g : \mathbb{F}_2^n \to \{-1,1\}$ is biased if $\mathbb{E}[g] \neq 0$). Using standard sampling methods we convert this to approximation using only a few biased $(k-1)$-iterated derivatives (Lemma 10). The final step is approximating each biased $(k-1)$-iterated derivative by a small number of its derivatives (which are k-iterated derivatives of f). To this end we prove a general lemma showing that any biased function can be approximated

in a concise manner by an algorithm having oracle access to a small number of its derivatives (Lemma 9).

We now apply the approximation by iterative derivative result we just described to bound the weight distribution of Reed-Muller codes. Fix $\delta = \delta(d)$ to be specified later. The approximation lemma implies that every $\mathrm{RM}(n,d)$ codeword of weight up to $2^{-k}(1-\epsilon)$ can be well approximated by a function of $c = c(k,\epsilon,\delta)$ of its k-iterated derivatives. Now we use the minimal distance of Reed-Muller codes. Any two distinct codewords $f', f'' \in \mathrm{RM}(n,d)$ have distance at least 2^{-d}. Thus, if we have a good enough approximation of f' (that is, for $\delta < 2^{-(d+1)}$), then such an approximation determines f' uniquely. Hence, to upper bound the number of Reed-Muller codewords it is enough to upper bound the number of δ-approximations for these codewords.

Using the approximation result we obtained, we get that the number of $\mathrm{RM}(n,d)$ codewords up to weight $2^{-k}(1-\epsilon)$, is bounded by the number of possible distinct inputs for the approximation function $\mathcal{A}$: the set of directions $\{y_{i,j}\}$ and the directional derivatives functions $f_{y_{1,1},\ldots,y_{1,k}}(x),\ldots,f_{y_{c,1},\ldots,y_{c,k}}(x)$. Each direction $y_{i,j}$ is an element of $\mathbb{F}_2^n$, hence has 2^n possible values; each k-iterated derivative is an n-variate polynomial of total degree at most $d-k$, hence has at most $2^{O(n^{d-k})}$ possible values. Thus, we get that the number of Reed-Muller codewords of weight up to $2^{-k}(1-\epsilon)$ can be bounded by

$$A(2^{-k}(1-\epsilon)) \leq 2^{n\cdot kc + O(n^{d-k})\cdot c}.$$

Combining these with the estimates we get for c we get the required upper bound. We complement these upper bound estimations with matching lower bounds. The bounds on the list-decoding size of Reed-Muller codes are obtained using similar techniques.

A similar work along the same lines is the work of Bogdanov and Viola [3], which shows that a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ whose weight is bounded by $\mathrm{wt}(f) \leq 1/2 - \epsilon$ can be well approximated by $c = c(k,\epsilon)$ of its 1^{st}-derivatives. Note that approximation by 1^{st}-derivatives *does not* imply in general approximation by k-iterated derivatives which is crucial for obtaining our bounds here.

1.6 Generalized Reed-Muller Codes

The problems of bounding both the accumulative weight distribution and the list-decoding size can be extended to Generalized Reed-Muller codes, the family of low-degree polynomials over larger fields. However, our techniques fail to prove tight results in these cases, as they do for Reed–Muller codes. We provide in Section 4 some partial results for this case and make a conjecture about the correct bounds.

1.7 Organization

The paper is organized as follows. In Section 2 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 3. We study the extension of our techniques for Generalized Reed-Muller codes in Section 4, where we provide some (non tight) bounds for these codes.

2 Approximation of biased functions by derivatives

We prove in this section the main technical lemma we use for bounding the weight distribution and list-decoding size of Reed–Muller codes. We require some definitions before stating it.

Definition 5 (Discrete derivatives)**.** Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ by a function. The discrete derivative of f in direction $a \in \mathbb{F}_2^n$ is defined as

$$f_a(x) = f(x+a) + f(x).$$

The k-iterated discrete derivative of f in directions $a_1, ..., a_k \in \mathbb{F}_2^n$ is defined as

$$\begin{aligned} f_{a_1,...,a_k}(x) =& (...((f_{a_1})_{a_2})...)_{a_k}(x) = \\ & \sum_{S \subseteq [k]} f(x + \sum_{i \in S} a_i) \end{aligned}$$

We note that usually derivatives are defined as $f_a(x) = f(x+a) - f(x)$, but since we are working over $\mathbb{F}_2$, we can ignore signs. Another notion central to our proof is that of a bias of a function.

Definition 6 (Bias)**.** The bias of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is

$$\begin{aligned} \mathrm{bias}(f) =& \mathbb{E}_{x \in \mathbb{F}_2^n}[(-1)^{f(x)}] = \\ & \mathbb{P}[f = 0] - \mathbb{P}[f = 1] = \\ & 1 - 2\mathrm{wt}(f) \end{aligned}$$

Our main lemma states that if f is a function with small weight, then it can be approximated by an algorithm having oracle access to a small number of its iterated derivatives. In the following when we assume an algorithm $\mathcal{A}$ receives as input a function $g(\cdot)$, we mean $\mathcal{A}$ has the ability to evaluate g on any input. One example is if $\mathcal{A}$ receives a representation of g in some canonical form (when g is a polynomial, $\mathcal{A}$ receives as input its list of coefficients).

Lemma 7. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\epsilon)$. For every error parameter $\delta > 0$ there exists a universal algorithm $\mathcal{A}$ (that is, independent of f) with the following properties. $\mathcal{A}$ has two types of inputs. The first is an input $x \in \mathbb{F}_2^n$ on which $\mathcal{A}$ is required to guess $f(x)$. The second input is a family of $t = O(\log(1/\epsilon\delta) \cdot \log(1/\delta))$ sets of k directions $\{y_{i,j} \in \mathbb{F}_2^n : 1 \leq i \leq t, 1 \leq j \leq k\}$ and their corresponding k-iterated derivatives of f, $\{f_{y_{i,1},...,y_{i,k}}(\cdot) : 1 \leq i \leq t\}$. For every function f there exists a set of directions $\{y_{i,j}\}$ such that*

$$\begin{aligned} \Pr_{x \in \mathbb{F}_2^n}[f(x) \neq \mathcal{A}(x; & \{y_{i,j} : 1 \leq i \leq t, 1 \leq j \leq k\}, \\ & \{f_{y_{i,1},...,y_{i,k}}(\cdot) : 1 \leq i \leq t\})] \leq \delta. \end{aligned}$$

Our starting point in the proof of Lemma 7 is the following lemma, which states that if a function f has weight less than $2^{-k}(1-\epsilon)$, then it can be computed exactly by a its iterated $(k-1)$-derivatives, and moreover each of theses derivatives is biased.

Lemma 8. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\epsilon)$ for integer $k \geq 2$. Then the function $(-1)^{f(x)} : \mathbb{F}_2^n \to \{-1, 1\}$ can be written as*

$$\begin{aligned} & (-1)^{f(x)} = \\ & \mathbb{E}_{a_1,...,a_{k-1} \in \mathbb{F}_2^n}[\alpha_{a_1,...,a_{k-1}}(-1)^{f_{a_1,...,a_{k-1}}(x)}] \end{aligned}$$

where

1. *The coefficients $\alpha_{a_1,...,a_{k-1}}$ are real numbers of absolute value at most 10.*
2. *All the functions $f_{a_1,...,a_{k-1}}$ are biased, $\mathit{bias}(f_{a_1,...,a_{k-1}}) \geq \epsilon$.*

We prove Lemma 8 in Subsection 2.1. The second lemma shows that biased functions can be approximated using a small number of their derivatives.

Lemma 9. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that* $\mathrm{bias}(f) \geq \epsilon$. *For every error parameter $\delta > 0$ there exists a universal algorithm $\mathcal{A}'$ (that is, independent of f) with the following properties. $\mathcal{A}'$ has two types of inputs. The first is an input $x \in \mathbb{F}_2^n$ on which $\mathcal{A}'$ is required to guess $f(x)$. The second input is a set of $t = \log(1/\epsilon\delta)+1$ directions $y_1, \ldots, y_t \in \mathbb{F}_2^n$ and the directional derivatives of f in these directions $f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)$. For every function f there exists a set of directions $y_1, \ldots, y_t$ such that*

$$\Pr_{x \in \mathbb{F}_2^n}\left[f(x) \neq \mathcal{A}'(x; y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot))\right] \leq \delta.$$

We prove Lemma 9 in Subsection 2.2. The last ingredient required for the proof of Lemma 7 is a standard sampling lemma showing how to transform exact computation by averaging many functions, to approximation by averaging few functions.

Lemma 10. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function, $H = \{h_1, ..., h_t\}$ a set of functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$, such that there exist constants $c_{h_1}, ..., c_{h_t}$ of absolute value at most C, such that*

$$(-1)^{f(x)} = \mathbb{E}_{i \in [t]}[c_{h_i}(-1)^{h_i(x)}] \qquad (\forall x \in \mathbb{F}_2^n)$$

Then f can be approximated by a small number of the functions $h_1, ..., h_t$. For any error parameter $\delta > 0$, there exist functions $h_1, ..., h_\ell \in H$ for $\ell = O(C^2 \log 1/\delta)$, and a function $F : \mathbb{F}_2^\ell \to \mathbb{F}_2$, such that the relative distance between $f(x)$ and $F(h_1(x), ..., h_\ell(x))$ is at most δ, i.e.

$$\mathbb{P}_{x \in \mathbb{F}_2^n}[f(x) \neq F(h_1(x), ..., h_\ell(x))] \leq \delta$$

The function F is a weighted majority, i.e. it is of the form

$$F(h_1(x), ..., h_\ell(x)) = \mathrm{sign}(\sum_{i=1}^{\ell} s_i(-1)^{h_i(x)}).$$

Moreover, we can have $s_1, ..., s_\ell$ to be integers of absolute value at most $C + 1$.

We prove Lemma 10 in Subsection 2.3. We now prove Lemma 7 using Lemmas 8, 9 and 10.

Proof of Lemma 7. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\epsilon)$, and let $\delta > 0$ be an error parameter. We start by defining an algorithm $\mathcal{A}_1(x)$ approximating $f(x)$ using a small number of its $(k-1)$-iterated derivatives. If $k = 1$ simply set $\mathcal{A}_1(x) = f(x)$. For $k \geq 2$ apply Lemma 8 to get that $(-1)^{f(x)}$ can be exactly computed as

$$\mathbb{E}_{a_1, \ldots, a_{k-1} \in \mathbb{F}_2^n}[\alpha_{a_1, \ldots, a_{k-1}}(-1)^{f_{a_1, \ldots, a_{k-1}}(x)}]$$

where $|\alpha_{a_1, \ldots, a_k}| \leq 10$ and $\mathrm{bias}(f_{a_1, \ldots, a_{k-1}}(x)) \geq \epsilon$. Applying Lemma 10 we get that f can be approximated by a small number of its $(k-1)$-iterated derivatives,

$$\Pr_{x \in \mathbb{F}_2^n}[\mathrm{Maj}(f_{a_{1,1}, \ldots, a_{1,k-1}}(x), \ldots, f_{a_{\ell,1}, \ldots, a_{\ell,k-1}}(x)) \neq f(x)] \leq \delta/2$$

where $\ell = O(\log 1/\delta)$. Define

$$\mathcal{A}_1(x) = \mathrm{Maj}(f_{a_{1,1}, \ldots, a_{1,k-1}}(x), \ldots, f_{a_{\ell,1}, \ldots, a_{\ell,k-1}}(x)).$$

We now approximate each $(k-1)$-iterated derivative by a small number of its derivatives. We will use Lemma 9 to this end. Notice this can be done since by Lemma 8 all $(k-1)$-iterated derivatives $f_{a_{i,1},\ldots,a_{i,k-1}}$ have bias of at least ϵ (and in the $k=1$ case, $\mathrm{bias}(f) \geq \epsilon$). Thus, for each $1 \leq i \leq \ell$ there exists $t = O(\log(1/\epsilon\delta))$ directions $y_{i,1}, \ldots, y_{i,t}$ such that

$$\begin{aligned}\Pr_{x\in\mathbb{F}_2^n}[&f_{a_{i,1},\ldots,a_{i,k-1}}(x) \neq \mathcal{A}'(x; y_{i,1}, \ldots, y_{i,k},\\ &f_{a_{i,1},\ldots,a_{i,k-1},y_{i,1}}(\cdot), \ldots, f_{a_{i,1},\ldots,a_{i,k-1},y_{i,t}}(\cdot))]\\ &\leq \delta/(2\ell).\end{aligned}$$

Plugging all these into $\mathcal{A}_1$, we get an algorithm $\mathcal{A}$ such that

$$\begin{aligned}\Pr_{x\in\mathbb{F}_2^n}[&f(x) \neq A(x; \{y_{i,j} : 1 \leq i \leq \ell, 1 \leq j \leq t\},\\ &\{f_{a_{i,1},\ldots,a_{i,k-1},y_{i,j}} : 1 \leq i \leq \ell, 1 \leq j \leq t\})] \leq \delta.\end{aligned}$$

In total, $\mathcal{A}$ has as input $\ell \cdot t = O(\log(1/\epsilon\delta) \cdot \log(1/\delta))$ k-iterated derivatives of f, and (a subset) of the directions of these derivatives. □

2.1 Proof of Lemma 8

Before proving Lemma 8, we need some claims regarding derivatives. The first claim shows that if a function has non-zero bias, it can be computed by an average of its derivatives.

Claim 11. *Let $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{bias}(g) \neq 0$. Then*

$$(-1)^{g(x)} = \frac{1}{\mathrm{bias}(g)}\mathbb{E}_{a\in\mathbb{F}_2^n}[(-1)^{g_a(x)}]$$

where the identity holds for any $x \in \mathbb{F}_2^n$.

Proof. Fix x. We have

$$\begin{aligned}&(-1)^{g(x)}\mathbb{E}_{a\in\mathbb{F}_2^n}[(-1)^{g_a(x)}] =\\ &\mathbb{E}_{a\in\mathbb{F}_2^n}[(-1)^{g(x)-g_a(x)}] =\\ &\mathbb{E}_{a\in\mathbb{F}_2^n}[(-1)^{g(x+a)}] = \mathrm{bias}(g)\end{aligned}$$

□

The following claim shows that if a function has low weight, then derivatives of it will also have low weight, and thus large bias.

Claim 12. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function such that $\mathrm{wt}(f) \leq 2^{-k}(1-\epsilon)$. Let $a_1, \ldots, a_s \in \mathbb{F}_2^n$ for $1 \leq s \leq k-1$ be any derivatives, and consider $\mathrm{bias}(f_{a_1,\ldots,a_s})$. Then $\mathrm{bias}(f_{a_1,\ldots,a_s}) \geq 1-2^{s+1-k}(1-\epsilon)$. In particular*

1. *If $s < k-1$ then $\mathrm{bias}(f_{a_1,\ldots,a_s}) \geq 1 - 2^{s+1-k}$.*
2. *If $s = k-1$ then $\mathrm{bias}(f_{a_1,\ldots,a_s}) \geq \epsilon$.*

Proof. Consider $f_{a_1,\ldots,a_s}$

$$f_{a_1,\ldots,a_s} = \sum_{I\subseteq[s]} f(x+\sum_{i\in I} a_i)$$

For random x, the probability that $f(x+\sum_{i\in I} a_i)=1$ is $\mathrm{wt}(f)$, which is at most $2^{-k}(1-\epsilon)$. Thus by the union bound,

$$\mathbb{P}_{x\in\mathbb{F}_2^n}[\exists I\subseteq[s],\ f(x+\sum_{i\in I} a_i)=1] \leq 2^{s-k}(1-\epsilon)$$

In particular it implies that

$$\mathrm{wt}(f_{a_1,\ldots,a_s}) = \mathbb{P}_{x\in\mathbb{F}_2^n}[f_{a_1,\ldots,a_s}(x)=1] \leq 2^{s-k}(1-\epsilon)$$

and we get the bound since $\mathrm{bias}(f_{a_1,\ldots,a_s}) = 1-2\mathrm{wt}(f_{a_1,\ldots,a_s})$. □

We now can prove Lemma 8 using Claims 11 and 12.

Proof of Lemma 8. Let $f:\mathbb{F}_2^n\to\mathbb{F}_2$ be a function such that $\mathrm{wt}(f)\leq 2^{-k}(1-\epsilon)$. Thus $\mathrm{bias}(f) = 1-2\mathrm{wt}(f) > 0$ and by Claim 11 we can write

$$(-1)^{f(x)} = \frac{1}{\mathrm{bias}(f)}\mathbb{E}_{a_1\in\mathbb{F}_2^n}[(-1)^{f_{a_1}(x)}]$$

If $k=1$ we are done. Otherwise by Claim 12, f_{a_1} also has positive bias,

$$\mathrm{bias}(f_{a_1}) \geq 1-2^{s+1-k}(1-\epsilon) > 0$$

and so again by Claim 11 we can write

$$(-1)^{f_{a_1}(x)} = \frac{1}{\mathrm{bias}(f_{a_1})}\mathbb{E}_{a_2\in\mathbb{F}_2^n}[(-1)^{f_{a_1,a_2}(x)}]$$

Thus we have

$$(-1)^{f(x)} = \frac{1}{\mathrm{bias}(f)}\mathbb{E}_{a_1\in\mathbb{F}_2^n}[\frac{1}{\mathrm{bias}(f_{a_1})}\mathbb{E}_{a_2\in\mathbb{F}_2^n}[(-1)^{f_{a_1,a_2}(x)}]]$$

We can continue this process as long as we can guarantee that $f_{a_1,\ldots,a_s}$ has non-zero bias for all $a_1,\ldots,a_s\in\mathbb{F}_2^n$. By Claim 12 we know this happens for $s\leq k-1$, and thus we have

$$(-1)^{f(x)} = \mathbb{E}_{a_1,\ldots,a_{k-1}\in\mathbb{F}_2^n}[\alpha_{a_1,\ldots,a_{k-1}}(-1)^{f_{a_1,\ldots,a_{k-1}}(x)}]$$

where

$$\alpha_{a_1,\ldots,a_k} = \frac{1}{\mathrm{bias}(f)}\frac{1}{\mathrm{bias}(f_{a_1})}\frac{1}{\mathrm{bias}(f_{a_1,a_2})}\cdots\frac{1}{\mathrm{bias}(f_{a_1,\ldots,a_{k-2}})}.$$

By Claim 12 we know that $\mathrm{bias}(f_{a_1,\ldots,a_{k-1}})\geq\epsilon$ for all $(k-1)$-iterated derivatives. We now bound $\alpha_{a_1,\ldots,a_k}$. By Claim 12 we get that

$$1\leq\alpha_{a_1,\ldots,a_k}\leq\prod_{s=0}^{k-2}\frac{1}{1-2^{s-k+1}}\leq\prod_{r\geq1}\frac{1}{1-2^{-r}}\leq 10.$$

□

2.2 Proof of Lemma 9.

For a set of directions $y_1, \ldots, y_t \in \mathbb{F}_2^n$ and a subset $I \subseteq [t]$, define $y_I = \sum_{i \in I} y_i$. We start by showing that if we know the directions $y_1, \ldots, y_t$ and the directional derivatives of f in these directions $f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)$, then we can also compute all the derivatives in directions y_I, that is the functions $f_{y_I}(\cdot)$.

Claim 13. *Let $y_1, \ldots, y_t \in \mathbb{F}_2^n$ a set of directions, and $f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)$ the directional derivatives of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$. For every non-empty $I \subseteq [t]$ there exists an algorithm $\mathcal{A}_I$ such that*

$$\mathcal{A}_I(x; y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)) = f_{y_I}(x)$$

for all $x \in \mathbb{F}_2^n$.

Proof. Let $I = \{i_1, \ldots, i_r\}$. The algorithm $\mathcal{A}_I$ calculates

$$\mathcal{A}_I(x) = \sum_{a=1}^{r} f_{y_{i_a}}(x + \sum_{b=1}^{a-1} y_{i_b}).$$

It is straightforward to verify that $\mathcal{A}_I(x) = f_{y_I}(x)$ for all $x \in \mathbb{F}_2^n$. □

We turn to prove Lemma 9.

Proof of Lemma 9. Define the algorithm $\mathcal{A}'$ as follows. For a set of directions $y_1, \ldots, y_t \in \mathbb{F}_2^n$ and the directional derivatives of $f : \mathbb{F}_2^n \to \mathbb{F}_2$ in these directions $f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)$, define $\mathcal{A}'(x)$ to be the majority vote of $f_{y_I}(x)$, which according to Claim 13 can be computed by algorithms receiving $x, y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)$, that is,

$$\begin{aligned}
&\mathcal{A}'(x; y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)) = \\
&\mathrm{Maj}\left\{\mathcal{A}_I(x; y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot))\right\}_{\emptyset \neq I \subseteq [t]} = \\
&\mathrm{Maj}\left\{f_{y_I}(x)\right\}_{\emptyset \neq I \subseteq [t]}.
\end{aligned}$$

We will prove that there is a choice of $y_1, \ldots, y_t$ for which $\mathcal{A}'(x) = f(x)$ for almost all x. In fact, we will prove this occurs for a random choice of $y_1, \ldots, y_t$. First, we claim that $\mathcal{A}'(x) = f(x)$ iff

$$S = \sum_{\emptyset \neq I \subseteq [t]} (-1)^{f(x+y_I)} > 0.$$

This is because $f_{y_I}(x) = f(x)$ iff $f(x + y_I) = 0$. Having the majority of $f_{y_I}(x)$ being equal to $f(x)$ is equivalent to $S > 0$ (note we cannot have $S = 0$ as S is the sum of an odd number of $\{-1, 1\}$ summands). Let $x, y_1, \ldots, y_t \in \mathbb{F}_2^n$ be chosen uniformly and independently. We prove $S > 0$ with high probability using Markov's inequality. First we compute $\mathbb{E}[S]$.

$$\mathbb{E}[S] = \mathbb{E}\left[\sum_{\emptyset \neq I \subseteq [t]} (-1)^{f(x+y_I)}\right] = (2^t - 1)\mathrm{bias}(f).$$

To bound $\mathrm{Var}[S]$ we observe that the different summands in S are pairwise independent. This is because for distinct $I, J \subseteq [t]$ we have

$$\begin{aligned}
&\mathbb{E}[(-1)^{f(x+y_I)} \cdot (-1)^{f(x+y_I)}] = \\
&\mathbb{E}[(-1)^{f(x+y_I)+f(x+y_J)}] = \\
&\mathbb{E}[(-1)^{f(x+y_I)}] \cdot \mathbb{E}[(-1)^{f(x+y_J)}] = \\
&bias(f)^2,
\end{aligned}$$

where we used the fact that the two points $x + y_I$ and $x + y_J$ are uniform and independent given that $x, y_1, \ldots, y_t$ are chosen uniformly and independently. We thus conclude that

$$\begin{aligned}\mathrm{Var}[S] &= \sum_{\emptyset \neq I \subseteq [t]} \mathrm{Var}[(-1)^{f(x+y_I)}] \\ &= (2^t - 1)\mathrm{Var}[(-1)^{f(x)}] \leq 2^t - 1.\end{aligned}$$

Hence we conclude that

$$\begin{aligned}\Pr[S \leq 0] &\leq \Pr[|S - \mathbb{E}[S]| \geq (2^t - 1)\mathrm{bias}(f)] \\ &\leq \frac{bias(f)}{2^t - 1}.\end{aligned}$$

Thus, for $t = \log(1/\epsilon\delta) + 1$ we get that

$$\Pr[S \leq 0] \leq \delta,$$

Hence we get that for uniformly chosen $x, y_1, \ldots, y_t$,

$$\begin{aligned}\Pr_{x, y_1, \ldots, y_t \in \mathbb{F}_2^n}[&\mathcal{A}'(x; y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)) \\ &\neq f(x)] \leq \delta.\end{aligned}$$

By an averaging argument, for every f there must exist a choice for $y_1, \ldots, y_t$ where

$$\Pr_{x \in \mathbb{F}_2^n}[\mathcal{A}'(x; y_1, \ldots, y_t, f_{y_1}(\cdot), \ldots, f_{y_t}(\cdot)) \neq f(x)] \leq \delta.$$

□

2.3 Proof of Lemma 10

The proof of Lemma 10 is based on a standard sampling argument.

Proof of Lemma 10. Choose $h_1, ..., h_\ell$ uniformly and independently from H. Fix $x \in \mathbb{F}_2^n$, and let Z_i be the random variable

$$Z_i = c_{h_i}(-1)^{h_i(x)}$$

and let $S = \frac{Z_1 + ... + Z_\ell}{\ell}$. We will use the fact that if $|S - (-1)^{f(x)}| < 1$ then $\mathrm{sign}(S) = (-1)^{f(x)}$. We first bound the probability that

$$|S - (-1)^{f(x)}| > 1/4$$

By regular Chernoff arguments for bounded independent variables, since $\mathbb{E}[S] = (-1)^{f(x)}$ and each Z_i is of absolute value of at most C, we get that

$$\mathbb{P}_{h_1, ..., h_\ell \in H}[|S - (-1)^{f(x)}| > 1/4] \leq e^{-\frac{\ell}{32C^2}}$$

(see for example Theorem A.1.16 in [2]).

In particular for $\ell = O(C^2 \log 1/\delta)$ we get that

$$\mathbb{P}_{h_1, ..., h_\ell \in H}[|S - (-1)^{f(x)}| > 1/4] \leq \delta$$

Thus by averaging arguments, there exists $h_1, ..., h_\ell$ such that

$$\mathbb{P}_{x\in\mathbb{F}_2^n}[|\frac{\sum_{i=1}^{\ell} c_{h_i}(-1)^{h_i(x)}}{\ell} - (-1)^{f(x)}| \geq 1/4] \leq \delta$$

We now round each coefficient to a close rational, without damaging the approximation error. The coefficient of $(-1)^{h_i(x)}$ is $\alpha_i = \frac{c_{h_i}}{\ell}$. If we round c_{h_i} to the closest integer $[c_{h_i}]$, we get that the coefficient of each $(-1)^{h_i(x)}$ is changed by at most $\frac{1}{2\ell}$, and thus the total approximation is changed by at most $1/2$. Hence we have

$$\mathbb{P}_{x\in\mathbb{F}_2^n}[|\frac{\sum_{i=1}^{\ell}[c_{h_i}](-1)^{h_i(x)}}{\ell}) - (-1)^{f(x)}| \geq 3/4] \leq \delta.$$

Thus we got that

$$\mathbb{P}_{x\in\mathbb{F}_2^n}[\text{sign}(\frac{\sum_{i=1}^{\ell}[c_{h_i}](-1)^{h_i(x)}}{\ell}) \neq (-1)^{f(x)}] \leq \delta.$$

Since dividing by ℓ does not change the sign we get

$$\mathbb{P}_{x\in\mathbb{F}_2^n}[\text{sign}(\sum_{i=1}^{\ell}[c_{h_i}](-1)^{h_i(x)}) \neq (-1)^{f(x)}] \leq \delta$$

□

3 Bounds for Reed-Muller codes

In this section we study the weight distribution and list-decoding size of Reed–Muller codes. Recall that $\text{RM}(n, d)$ denotes the code of multivariate polynomials $p(x_1, ..., x_n)$ over $\mathbb{F}_2$ of total degree at most d. In the following n and d will always stand for the number of variables and the total degree. We assume that $d \ll n$, and study in particular the case of constant d.

3.1 Weight distribution of Reed-Muller codes

We prove in this subsection our first main theorem, Theorem 4, which gives the asymptotic behavior of the weight distribution of Reed-Muller codes. It is a direct corollary of Theorem 14, giving an upper bound on the accumulative weight at distance $2^{\ell-d} - \epsilon$, and Lemma 15, giving a simple lower bound at distance $2^{\ell-d-1}$.

Theorem 14 (Upper bound on the accumulative weight)**.** *For any integer $1 \leq k \leq d-1$,*

$$A(2^{-k}(1-\epsilon)) \leq (1/\epsilon)^{O(\frac{d^2}{(d-k)!}n^{d-k})}.$$

In particular for constant d we get that

$$A(2^{-k} - \epsilon) \leq (1/\epsilon)^{O(n^{d-k})}.$$

Lemma 15 (Lower bound on the accumulative weight)**.** *For any integer $1 \leq k \leq d$*

$$A(2^{-k}) \geq 2^{\frac{n^{d-k+1}}{(d-k+1)!}(1+o(1))}.$$

In particular for constant d we get that

$$A(2^{-k}) \geq 2^{\Omega(n^{d-k+1})}.$$

We start by proving the lower bound.

Proof of Lemma 15. Single out k variables $x_1, ..., x_k$, and let q be any degree $d-k+1$ polynomials on the remaining $n-k$ variables. First, for any such q, the following degree d polynomial has relative weight exactly 2^{-k}

$$q'(x_1, ..., x_n) = x_1 x_2 ... x_{k-1}(x_k + q(x_{k+1}, ..., x_n))$$

The number of different polynomials q is

$$2^{\binom{n-k}{d-k+1}} = 2^{\frac{n^{d-k+1}}{(d-k+1)!}(1+o(1))}$$

□

We prove Theorem 14 using Lemma 7.

Proof of Theorem 14. Fix $1 \leq k \leq d-1$. We will bound the number of polynomials $p \in \mathrm{RM}(n,d)$ such that $\mathrm{wt}(p) \leq 2^{-k}(1-\epsilon)$. Let p be any such polynomial. Apply Lemma 7 to $p(x)$ with error parameter $\delta = 2^{-(d+2)}$. There exists a universal algorithm $\mathcal{A}$, and for each p a set of $t = O(d^2 + d\log(1/\epsilon))$ directions $\{y_{i,j} : 1 \leq i \leq t, 1 \leq j \leq k\}$ such that

$$\Pr_{x \in \mathbb{F}_2^n}[p(x) \neq \mathcal{A}(x; \{y_{i,j} : 1 \leq i \leq t, 1 \leq j \leq k\}, \\ \{p_{y_{i,1},...,y_{i,k}}(\cdot) : 1 \leq i \leq t\})] \leq \delta.$$

Define $p'(x) = \mathcal{A}\left(x; \{y_{i,j}\}, \{p_{y_{i,1},...,y_{i,k}}(\cdot)\}\right)$. We have that $\mathrm{dist}(p,p') = \Pr_x[p(x) \neq p'(x)] \leq \delta$. We claim that this guarantees that $p'(x)$ specifies $p(x)$ uniquely - it is the only element of $\mathrm{RM}(n,d)$ of distance at most δ from p'. This is because the minimal distance of $\mathrm{RM}(n,d)$ is 2^{-d}, and we chose δ to be less than half the minimal distance. Now, in order to compute $p'(x)$, we need to specify to the algorithm $\mathcal{A}$ the set of vectors $y_{i,j}$ and the polynomials $p_{y_{i,1},...,y_{i,k}}(\cdot)$. To specify each vector $y_{i,j} \in \mathbb{F}_2^n$ we require n bits. Each polynomial $p_{y_{i,1},...,y_{i,k}}(\cdot)$ is a k-iterated derivative of a degree-d polynomial $p(x)$, hence it is a degree $d-k$ polynomial. Thus, in order to specify it, we need to give the list of its $\sum_{i=0}^{d-k}\binom{n}{i}$ bits. Summing up, we need a total of

$$tk \cdot n + t \cdot \sum_{i=0}^{d-k}\binom{n}{i} = O\left(d^2\log(1/\epsilon) \cdot \frac{n^{d-k}}{(d-k)!}\right)$$

bits in order to specify p' completely. Since each p' approximates at most a single p we get that the number of polynomials $p \in \mathrm{RM}(n,d)$ such that $\mathrm{wt}(p) \leq 2^{-k}(1-\epsilon)$ is bounded by the number of distinct p', which is bounded by

$$(1/\epsilon)^{O\left(\frac{d^2}{(d-k)!}n^{d-k}\right)}.$$

□

3.2 List-decoding size of Reed-Muller codes

We now turn to the problem of bounding the list-decoding size of Reed-Muller codes, and we prove our second main theorem, Theorem 5. We will show that the same techniques used to bound the weight distribution when proving Theorem 4 can be applied with minor variants to also bound the list-decoding size. We note this is an exception; commonly bounding the list-decoding size is a

much harder task than bounding the weight distribution, and there exist codes where these two parameters behave very differently. However, we will see that in the case of Reed–Muller codes they share the same asymptotic behavior.

Theorem 5 giving the list-decoding size of Reed-Muller codes is a direct corollary of Theorem 16, giving an upper bound on the list-decoding size at distance $2^{\ell-d}-\epsilon$, and the same lower bound we used to bound the accumulative weight distribution, obtained in Lemma 15.

Theorem 16 (Upper bound on the list-decoding size)**.** *For any integer $1 \leq k \leq d-1$,*

$$L(2^{-k}(1-\epsilon)) \leq (1/\epsilon)^{O(\frac{d^2}{(d-k)!}n^{d-k})}.$$

In particular for constant d we get that

$$L(2^{-k}-\epsilon) \leq (1/\epsilon)^{O(n^{d-k})}.$$

Proof of Theorem 16. The proof follows the same lines as that of Theorem 14. Fix $f : \mathbb{F}_2^n \to \mathbb{F}_2$ to be any function. We will bound the number of polynomials $p \in \mathrm{RM}(n,d)$ such that $\mathrm{dist}(p,f) \leq 2^{-k}(1-\epsilon)$. Let $g = p + f$ such that $\mathrm{wt}(g) \leq 2^{-k}(1-\epsilon)$. Applying Lemma 7 to $g(x)$ with the error parameter $\delta = 2^{-(d+2)}$, there exists a universal algorithm $\mathcal{A}$ and a set of direction $\{y_{i,j} : 1 \leq i \leq t, 1 \leq j \leq k\}$ such that

$$\Pr_{x\in\mathbb{F}_2^n}[g(x) \neq \mathcal{A}(x;\{y_{i,j} : 1 \leq i \leq t, 1 \leq j \leq k\}, \{g_{y_{i,1},\ldots,y_{i,k}}(\cdot) : 1 \leq i \leq t\})] \leq \delta.$$

Since $g(x) = p(x) + f(x)$ we also have $g_{y_{i,1},\ldots,y_{i,k}}(\cdot) = p_{y_{i,1},\ldots,y_{i,k}}(\cdot) + f_{y_{i,1},\ldots,y_{i,k}}(\cdot)$. Hence, we can replace each instance of g or its derivatives in $\mathcal{A}$ with instances of p, f and their derivatives. Thus we get that

$$\Pr_{x\in\mathbb{F}_2^n}[p(x) \neq f(x) + \mathcal{A}(x;\{y_{i,j}\}, \{p_{y_{i,1},\ldots,y_{i,k}}(\cdot) + f_{y_{i,1},\ldots,y_{i,k}}(\cdot)\})] \leq \delta.$$

Define $p'(x) = f(x) + \mathcal{A}(x;\{y_{i,j}\},\{p_{y_{i,1},\ldots,y_{i,k}}(\cdot) + f_{y_{i,1},\ldots,y_{i,k}}(\cdot)\})$. Since we again have $\mathrm{dist}(p,p') \leq \delta$, the function $p'(x)$ specifies $p(x)$ uniquely as the only element in $\mathrm{RM}(n,d)$ which has distance at most δ from p'. Now, in order to compute p', we may assume the algorithm $\mathcal{A}$ has oracle access to the function $f(\cdot)$, since we have fixed it in advance, and it is the same for all the polynomials we wish to bound. Thus, in order to calculate $p'(x)$, we need to provide to the algorithm $\mathcal{A}$ the set of directions $y_{i,j}$ and the polynomials $p_{y_{i,1},\ldots,y_{i,k}}(\cdot)$. Notice that $\mathcal{A}$ can compute $f_{y_{i,1},\ldots,y_{i,k}}(\cdot)$ using the oracle access to f and the set of directions $y_{i,j}$. As in the proof of Theorem 14, each direction $y_{i,j} \in \mathbb{F}_2^n$ requires n bits, and each polynomial $p_{y_{i,1},\ldots,y_{i,k}}(\cdot)$ being a degree $d-k$ polynomial requires $\sum_{i=0}^{d-k}\binom{n}{i}$ bits to specify. Following the same calculations as those in the proof of Theorem 14, we conclude that the number of distinct $p'(x)$ is bounded by

$$(1/\epsilon)^{O(\frac{d^2}{(d-k)!}n^{d-k})}.$$

Thus, for every fixed function f, this is also a bound on the number of $p \in \mathrm{RM}(n,d)$ such that $\mathrm{dist}(p,f) \leq 2^{-k}(1-\epsilon)$. □

4 Generalized Reed-Muller codes

The problems of bounding both the accumulative 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. We briefly describe the reasons below, and give some partial results.

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 $p(x_1,...,x_n)$ over the field $\mathbb{F}_q$, of total degree at most d.

Definition 7. 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 8. The relative distance between two functions $f,g:\mathbb{F}_q^n \to \mathbb{F}_q$ is defined as

$$\mathrm{dist}(f,g) = \mathbb{P}_{x\in\mathbb{F}_q^n}[f(x) \neq g(x)]$$

The accumulative 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 A_q and L_q. For each $1 \leq k \leq d$, we define a distance r_k as follows.

1. For $k=1$, let $d=(q-1)a+b$, where $1 \leq b \leq q-1$. Define $r_1 = q^{-a}(1-b/q)$. r_1 is the minimal distance of $\mathrm{GRM}_q(n,d)$.
2. For $2 \leq k \leq d-1$, let $d-k=(q-1)a+b$, where $1 \leq b \leq q-1$. 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 accumulative weight distribution and the list-decoding size, the distances r_k are the thresholds for the exponential dependency in n.

Conjecture 17. *Let $\epsilon > 0$ be constant, and consider $\mathrm{GRM}_q(n,d)$ for constant d. Then*

- *For $\alpha \leq r_1 - \epsilon$ both $A_q(\alpha)$ and $L_q(\alpha)$ are constants.*
- *For $r_k \leq \alpha \leq r_{k+1} - \epsilon$ both $A_q(\alpha)$ and $L_q(\alpha)$ are $2^{\Theta(n^k)}$.*
- *For $\alpha \geq r_d$ both $A_q(\alpha)$ and $L_q(\alpha)$ are $2^{\Theta(n^d)}$.*

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

Lemma 18 (Lower bound for A_q)**.** *For any integer $1 \leq k \leq d$,*

$$A_q(r_k) \geq 2^{\Omega(n^k)}$$

The problem is proving matching upper bounds. 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 $q > 2$.

$$A_q(2^{-k} - \epsilon) \leq 2^{O(n^{d-k})}$$

If we would like to get upper bounds closer to the lower bounds, a natural approach would be to generalize Lemma 8 to taking several derivatives in the same direction (which is possible over

larger fields). This would give tight results for some values of k. The crucial point is generalizing Claim 11 to the case of taking multiple derivatives in the same direction. So far, we didn't find a way of doing so.

Instead, we give partial results for Conjecture 17 at both ends of the spectrum. We give results when $\alpha \leq r_1 - \epsilon$, and when $r_{d-1} \leq \alpha \leq r_d - \epsilon$ (when $\alpha \geq r_d$ Lemma 18 gives $L_q(\alpha)$ and $A_q(\alpha)$ 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 $\epsilon > 0$, $A_q(r_1 - \epsilon) = 1$. Gopalan, Klivans and Zuckerman [7] prove that $L_q(r_1 - \epsilon)$ is constant when $q-1$ divides d.

Theorem 19 (Corollary 18 in [7])**.** *Assume $q-1$ divides d. Then*

$$L_q(r_1 - \epsilon) \leq c(q, d, \epsilon)$$

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

Lemma 20. *Let $\epsilon > 0$ be constant. then*

$$A_q(r_d - \epsilon) \leq 2^{O(n^{d-1})}$$

We now move on to prove Lemmas 18 and 20. We start with Lemma 18.

Proof of Lemma 18. We start by proving for $2 \leq k \leq d-1$. Let $d - k = (q-1)a + b$, where $1 \leq b \leq q-1$. Single out $a+2$ variables $x_1, ..., x_{a+2}$, and let g be any degree k polynomial on the remaining variables. The following polynomial has degree d and weight exactly $q^{-a}(1-b/q)(1-1/q)$.

$$\begin{aligned} &g'(x_1, ..., x_n) = \\ &\left(\prod_{i=1}^{a}\prod_{j=1}^{q-1}(x_i - j)\right) \cdot \left(\prod_{j=1}^{b}(x_{a+1} - j)\right) \cdot \\ &(x_{a+2} + g(x_{a+3}, ..., x_n)) \end{aligned}$$

The number of distinct polynomial g 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 $l_1(x), ..., l_{a+1}(x)$ be any independent linear functions, and consider

$$\begin{aligned} &g'(x_1, ..., x_n) = \\ &\left(\prod_{i=1}^{a}\prod_{j=1}^{q-1}(l_i(x) - j)\right)\left(\prod_{j=1}^{b}(l_{a+1}(x) - j)\right) \end{aligned}$$

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

We now continue to prove Lemma 20. We first make some necessary definitions.

Definition 9. The bias of a polynomial $p(x_1, ..., x_n)$ over $\mathbb{F}_q$ is defined to be

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

where $\omega = e^{2\pi i/q}$ is a primitive q-th 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 21 (Theorem 2 in [9]). *Let $p(x_1, ..., x_n)$ be a degree d polynomial, such that $|\text{bias}(p)| \geq \epsilon$. Then p can be decomposed as a function of a constant number of lower degree polynomials*

$$p(x) = F(g_1(x), ..., g_c(x))$$

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

We will use Theorem 21 to bound $A(r_d - \epsilon)$ for any constant $\epsilon > 0$.

Proof of Lemma 20. We will show that any polynomial $p \in \text{GRM}_q(n, d)$ such that $\text{wt}(p) \leq 1 - 1/p - \epsilon$ can be decomposed as

$$p(x) = F(g_1(x), ..., g_c(x))$$

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 p be such that $\text{wt}(p) \leq 1 - 1/p - \epsilon$. We will show there exists $\alpha \in \mathbb{F}_q$, $\alpha \neq 0$ such that $\text{bias}(\alpha p) \geq \epsilon$. We will then finish by using Theorem 21 on the polynomial αp.

Consider the bias of αp for random $\alpha \in \mathbb{F}_q$.

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

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

$$\mathbb{E}_{\alpha \in \mathbb{F}_q \setminus \{0\}}[\text{bias}(\alpha p)] = 1 - \frac{q}{q-1}\text{wt}(p) \geq \frac{q}{q-1}\epsilon$$

So, there must exist $\alpha \neq 0$ such that $\text{bias}(\alpha p) \geq \epsilon$. □

Acknowledgement. We would like to thank Madhu Sudan for helpful comments on this work. The second author would like to thank his advisor, Omer Reingold, for on-going advice and encouragement. He would also like to thank Microsoft Research for their support during his internship. The first author was supported in part by NSF Awards CCF-0514167 and NSF-0729011. The second author was supported partly by the Israel Science Foundation (grant 1300/05). Research was conducted partly when the second author was an intern at Microsoft Research.

References

[1] S. Azumi and T. Kasami and N. Tokura, *On the Weight Enumeration of Weights Less than 2.5d of Reed-Muller Codes*, Information and Control, 30(4): 380–395, 1976.

[2] N. Alon and J. Spencer, *The Probabilistic Method*, Second edition, published by John Wiley, 2000.

[3] A. Bogdanov and E. Viola. *Pseudorandom bits for polynomials via the Gowers norm.* In the 48th Annual Symposium on Foundations of Computer Science (FOCS 2007).

[4] O. Goldreich and L. Levin, *A hard core predicate for all one way functions*, In the Proceedings of the 21st ACM Symposium on Theory of Computing (STOC), 1989.

[5] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan, *Learning polynomials with queries: The highly noisy case*, SIAM Journal on Discrete Mathematics, 13(4):535-570, November 2000.

[6] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan, *Learning polynomials with queries: The highly noisy case*, SIAM Journal on Discrete Mathematics, 13(4):535-570, November 2000.

[7] P. Gopalan, A. Klivans and D. Zuckerman, *List-Decoding Reed Muller Codes over Small Fields*, In the Proceedings of the 40th ACM Symposium on Theory of Computing (STOC), 2008.

[8] V. Guruswami, *List decoding of Error-Correcting Codes*, vol 3282 of Lecture notes in Computer Science, Springer 2004.

[9] T. Kaufman and S. Lovett, *Worst case to Average Case Reductions for Polynomials*, To appear in the Proceedings of the 49th Annual Symposium on Foundations of Computer Science (FOCS), 2008.

[10] E. Kushilevitz and Y. Mansour, *Learning Decision Trees using the Fourier Spectrum*, SIAM Journal of Computing, 22(6), (1993), pp 1331-1348.

[11] T. Kasami and N. Tokura, *On the weight structure of Reed-Muller codes*, In the IEEE Transactions on Information Theory 16 (Issue 6), 1970.

[12] J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, Amsterdam, North-Holland, 1977.

[13] M. Sudan, *Decoding of Reed-Solomon codes beyond the error-correction bound*, Journal of Complexity, 13, (1997), pp. 180-193.

[14] M. Sudan, *List decoding: Algorithms and Applications* , SIGACT News, 31 (2000), pp 16-27.

[15] M. Sudan, L. Trevian, S. Vadhan *Pseudorandom Generators without the XOR Lemma*, J. Comput. Syst. Sci., 61 (2001), pp 236-266.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de