Reed-Muller Codes for Random Erasures and Errors

[Extended Abstract] *

Emmanuel Abbe
PACM & EE Department
Princeton University
212 Fine Hall
Princeton, NJ 08540, USA
eabbe@princeton.edu

Amir Shpilka
CS Department
Tel-Aviv University
Schreiber 118
Tel-Aviv, 69978, Israel
shpilka@post.tau.ac.il

Avi Wigderson
School of Mathematics
Institute for Advanced Study
1 Einstein Drive
Princeton, NJ 08540, USA
avi@ias.edu

ABSTRACT

This paper studies the parameters for which binary Reed-Muller (RM) codes can be decoded successfully on the BEC and BSC, and in particular when can they achieve capacity for these two classical channels. Necessarily, the paper also studies properties of evaluations of multi-variate $GF(2)$ polynomials on random sets of inputs.

For erasures, we prove that RM codes achieve capacity both for very high rate and very low rate regimes. For errors, we prove that RM codes achieve capacity for very low rate regimes, and for very high rates, we show that they can uniquely decode at about square root of the number of errors at capacity.

The proofs of these four results are based on different techniques, which we find interesting in their own right. In particular, we study the following questions about $E(m, r)$, the matrix whose rows are truth tables of all monomials of degree $\leq r$ in $m$ variables. What is the most (resp. least) number of random columns in $E(m, r)$ that define a submatrix having full column rank (resp. full row rank) with high probability? We obtain tight bounds for very small (resp. very large) degrees $r$, which we use to show that RM codes achieve capacity for erasures in these regimes.

Our decoding from random errors follows from the following novel reduction. For every linear code $C$ of sufficiently high rate we construct a new code $C'$ obtained by tensoring $C$, such that for every subset $S$ of coordinates, if $C$ can recover from erasures in $S$, then $C'$ can recover from errors in $S$. Specializing this to RM codes and using our results for erasures imply our result on unique decoding of RM codes at high rate.

Finally, two of our capacity achieving results require tight bounds on the weight distribution of RM codes. We obtain such bounds extending the recent [27] bounds from constant degree to linear degree polynomials.

*See [2] for a full version of this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

STOC’15, June 14–17, 2015, Portland, Oregon, USA.
Copyright © 2015 ACM 978-1-4503-3536-2/15/06 ...$15.00.
http://dx.doi.org/10.1145/2746539.2746575.

Categories and Subject Descriptors

E.4 [Coding and Information Theory]: Error control codes

General Terms

Theory

Keywords

Reed-Muller codes; channel capacity; weight enumerator; multivariate polynomials.

1. INTRODUCTION

1.1 Overview

We start by giving a high level description of the background and motivation for the problems we study, and of our results.

Reed-Muller (RM) codes were introduced in 1954, first by Muller [33] and shortly after by Reed [36], who also provided a decoding algorithm. They are among the oldest and simplest codes to construct; the codewords are the evaluation vectors of all multivariate polynomials of a given degree bound. More precisely, in an $RM(m, r)$ code over a finite field $\mathbb{F}$, a message is interpreted as the coefficients of a multivariate polynomial $f$ of degree at most $r$ over $\mathbb{F}$, and its encoding is simply the vector of evaluations $f(a)$ for all possible assignments $a \in \mathbb{F}^m$ to the variables. Thus, RM codes are linear codes. They have been extensively studied in coding theory, and yet some of their most basic coding-theoretic parameters remain a mystery to date. Specifically, fixing the rate of an RM code, while it is easy to compute its tolerance to errors and erasures in the worst-case (or adversarial) model, it has proved extremely difficult to estimate this tolerance for even the simplest models of random errors and erasures. The questions regarding erasures can be interpreted from a learning theory perspective, about interpolating low degree polynomials from lossy or noisy evaluations. The questions regarding errors relate sparse recovery from random Boolean errors. This paper makes some progress on these basic questions.

Reed-Muller codes (over both large and small finite fields) have been extremely influential in the theory of computation, playing a central role in some important developments in several areas. In cryptography, they have been used e.g. in secret sharing schemes [37], instance hiding constructions [10] and private information retrieval (see the survey
In the theory of randomness, they have been used in the constructions of many pseudo-random generators and randomness extractors, e.g.,[12]. These in turn were used for hardness amplification, program testing and eventually in various interactive and probabilistic proof systems, e.g. the celebrated results NEXP=MP [8], IP=PSPACE [38], NP=PCP [6]. In circuit lower bounds for some low complexity classes one argues that every circuit in the class is close to a codeword, so any function far from the code cannot be computed by such circuits (e.g. [35]. In distributed computing they were used to design fault-tolerant information dispersal algorithms for networks [34]. The hardness of approximation of many optimization problems is greatly improved by the “short code” [9], which uses the optimal testing result of [11]. And the list goes on. Needless to say, the properties used in these works are properties of low-degree polynomials (such interpolation, linearity, partial derivatives, self-reducibility, heredity under various restrictions to variables, etc.), and in some of these cases, specific coding-theoretic perspective such as distance, unique-decoding, list-decoding, local testing and decoding etc. play important roles. Finally, polynomials are basic objects to understand computationally from many perspectives (e.g. testing identities, factoring, learning, etc.), and this study interacts well with the study of coding theoretic questions regarding RM codes.

To discuss the coding-theoretic questions we focus on, and give appropriate perspective, we need some more notation. First, we will restrict attention to binary codes, the most basic case where $\mathbb{F} = \mathbb{F}_2$, the field of two elements$^1$. To reliably transmit $k$-bit messages we encode each by an $n$-bit codeword via a mapping $C : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n$. We abuse notation and denote by $C$ both the mapping and its image, i.e., the set of all codewords$^2$. The rate of $C$ is given by the ratio $k/n$, capturing the redundancy of the code (the smaller it is, the more redundant it is). A major problem of coding theory is to determine the largest rate for which one can uniquely recover the original message from a corrupted codeword (naturally, explicit codes with efficient encoding and decoding algorithms are desirable). This of course depends on the nature of corruption, and we shall deal here with the two most basic ones, erasures (bit-losses) and errors (bit-flips). Curiously, the two seminal papers from the late 1940s giving birth to coding theory, by Shannon [39] and Hamming [22] differ in whether one should consider recovery of most corruptions, of from all corruptions. In other words, Shannon advocates average-case analysis whereas Hamming advocates worst-case analysis.

In Hamming’s worst case setting, recovery of the original message must be possible from every corruption of every codeword. In this model there is a single parameter of the code determining recoverability: the distance of the code. The distance $d$ is the minimum Hamming distance of any two codewords in $C$ (the relative distance is simply the distance normalized by the block-length $n$). If the distance is $d$, then we can uniquely recover from at most $d$ erasures and from $(d-1)/2$ errors. This leaves the problem of finding the optimal trade-off between rate and distance, and designing codes which achieve this optimum. While these are still difficult open problems, we know a variety of codes that can simultaneously achieve constant rate and constant relative distance (such codes are often called asymptotically good). In contrast, Reed-Muller codes fall far short of that. The rate of $RM(m,r)$ is $(\binom{m}{r})/2^m$, while the distance is easily seen to be $2^{m-r}$. Thus making any one of them a positive constant makes the other exponentially small in $n$. In short, from a worst-case perspective, RM codes are pretty bad.

In Shannon’s average-case setting (which we study here), a codeword is subjected to a random corruption, from which recovery should be possible with high probability. This random corruption model is called a channel, and the best achievable rate is called the capacity of the channel. The two most basic ones, the Binary Erasure Channel (BEC) and the Binary Symmetric Channel (BSC), have a parameter $p$ (which may depend on $n$), and corrupt a message by independently replacing, with probability $p$, the symbol in each coordinate, with a “lost” symbol in the BEC($p$) channel, and with the complementary symbol in the BSC($p$) case. Shannon’s original paper already contains the optimal trade-off achievable for these (and many other channels). For every $p$, the capacity of BEC($p$) is $1-p$, and the capacity of BSC($p$) is $1-h(p)$, where $h$ is the binary entropy function. While Shannon shows that random codes achieve this optimal behavior,$^3$ explicit and efficiently encodable and decodable codes achieving capacity in both channels$^4$ have been obtained [18], among which are the recent Polar Codes [5] that we shall soon discuss.

Do Reed-Muller codes achieve capacity for these natural channels (despite their poor rate-distance trade-off)? The coding theory community seems to believe the answer is positive, and conjectures to that effect were made$^5$ in [13, 4, 32]. However, to date, we do not know any value of $p$ for which RM codes achieve the capacity for erasures or errors! This paper provides the first progress on this conjecture, resolving it for very low rates and very high rates (i.e., for polynomials of degrees $r$ which are very small or very large compared to the number of variables $m$). Our results unfortunately fall short of approximating the cases where the corruption rate $p$ is a constant, the most popular regime in coding theory.

The conjecture that RM codes achieve capacity has been experimentally “confirmed” in simulations [4, 32]. Moreover, despite being extremely old, new interest in it resurfaced a few years ago with the advent of polar codes [5]. To explain the connection between the two, as well as some of the technical problems arising in proving the results above, consider the following $2^m \times 2^m$ matrix $E_m$ (for “evaluation”). Index the rows and columns by all possible $m$-bit vectors in $\mathbb{F}_2^m$ in lexicographic order. Interpret the columns simply as points in $\mathbb{F}_2^m$, and the rows as monomials (where an $m$-bit string correspond to the monomial which is the product of variables in the positions containing a 1). Finally, $E_m(x,y)$ is the value of the monomial $x$ on the point $y$ (i.e., it is 1 if the set of 1’s in $x$ is contained in the set of 1’s in $y$). Thus, every row of $E_m$ is the truth table of one monomial. It is thus easy

$^1$The fact that random linear codes can achieve capacity for symmetric channels was shown in [17].

$^2$A code is linear if the mapping $C$ is $\mathbb{F}_2$-linear, or equivalently if the set of codewords $C$ is a linear subspace of $\mathbb{F}_2^m$.

$^3$For the case of the BEC, [30] provides the first LDPC codes that are capacity-achieving, and further LDPC ensembles have been recently developed with spatial coupling [29].

$^4$The belief that RM codes achieve capacity seems to be much older.
to see that the code \( R(m, r) \) is simply the span of the top (or “high weight”) \( k \) rows of \( E_m \), with \( k = \binom{n}{r} \); these are the truth tables of all degree \( \leq r \) polynomials. In contrast, polar codes of the same rate are spanned by a different set of \( k \) rows, so they form a different subspace of polynomials. While the monomials indexing the polar code rows have no explicit description (so far), they can be computed efficiently for any \( k \) in \( \text{poly}(n) = 2^{O(n)} \) time. It is somehow intuitively “better” to prefer higher weight rows to lower weight ones as the basis of the code (as the “chances of catching an error” seem higher). Given the amazing result that polar codes achieve capacity, this intuition seems to suggest that RM codes do so as well. In fact, experimental results in [32] suggest that RM codes may outperform polar codes for the BEC, this requires inverting a matrix over \( \text{GF}(2) \) to see that the code \( R \) suggests that RM codes may outperform polar codes for the BEC, this requires inverting a matrix over \( \text{GF}(2) \) to see that the code \( R \) achieves capacity, this intuition seems to suggest that RM codes do so as well. In fact, experimental results in [32] suggest that RM codes may outperform polar codes for the BEC, this requires inverting a matrix over \( \text{GF}(2) \) to see that the code \( R \) achieves capacity, this intuition seems to suggest that RM codes do so as well. In fact, experimental results in [32] suggest that RM codes may outperform polar codes for the BEC, this requires inverting a matrix over \( \text{GF}(2) \) to see that the code \( R \) achieves capacity. 

Denoting by \( E(m, r) \) the top submatrix of \( E_m \) with \( k = \binom{n}{r} \) rows, one can express some natural problems concerning it which are essential for our results. To obtain some of our results on achieving capacity for the erasure channel, we must understand the following two natural questions regarding \( E(m, r) \). First, what is the largest number \( s \) so that \( s \) random columns of \( E(m, r) \) are linearly independent with high probability. Second, what is the smallest number \( t \) such that \( t \) random columns have full row-rank. Capacity achieving for erasures means that \( s = (1 - o(1))k \) and \( t = (1 + o(1))k \), respectively. We prove that this is the case for small values of \( r \). The second property gives directly the result for low-rate codes \( R(m, r) \), and the first implies the result for high-rate codes using a duality property of RM codes. Both results may be viewed from a learning theory perspective, showing that in these ranges of parameters any degree \( r \) polynomial in \( m \) variables can be uniquely interpolated with high probability from its values on the minimum possible number of random inputs.

For errors, further analysis is needed beyond the rank properties discussed above. From the parity-check matrix viewpoint, decoding errors is equivalent to solving (with high probability) an underdetermined system of equations. Recall that a linear code can be expressed as the null space of an \((n - k) \times n\) parity-check matrix \( H \). If \( Z \) is a random error vector with about \((1 + o(1))k \) errors, then \( \text{poly}(n) = 2^{O(n)} \) time. It is somehow intuitively “better” to prefer higher weight rows to lower weight ones as the basis of the code (as the “chances of catching an error” seem higher). Given the amazing result that polar codes achieve capacity, this intuition seems to suggest that RM codes do so as well. In fact, experimental results in [32] suggest that RM codes may outperform polar codes for the BEC, this requires inverting a matrix over \( \text{GF}(2) \) to see that the code \( R \) achieves capacity. 

\[ u \in \left\{ 0, 1 \right\}^n \]

\[ \text{Hamming weight of } x \in \mathbb{F}_2^n \text{ is denoted } w(x) = \left| \{ i \in [n] : x_i \neq 0 \} \right| \text{ and the relative weight is } wt(x) = w(x)/n. \]

\[ B(n, s) = \{ x \in \mathbb{F}_2^n : w(x) \leq s \} \]

\[ \partial B(n, s) = \{ x \in \mathbb{F}_2^n : w(x) = s \}. \]

\[ \text{We use } \binom{n}{s} \text{ to denote the set of subsets of } [n] \text{ of cardinality } s. \]

\[ \text{For a vector } x \text{ of dimension } n \text{ and subset } S \text{ of } n, \text{ we use } x[S] \text{ to denote the components of } x \text{ indexed by } S, \text{ and if } X \text{ is matrix with } n \text{ columns, we use } X[S] \text{ to denote the subset of columns indexed by } S. \]

\[ E(m, r)[U] = U'. \]

When we need to be more explicit, for an \( a \times b \) matrix \( A \) and

\[ \text{known about the weight distribution of Reed-Muller code until the recent breakthrough paper of [27], which gave nearly tight bounds for constant degree polynomials for both. The results of [27] also apply to list-decoding of RM codes, which was previously investigated in [20]. We need a sharpening of their upper bound for two of our results, which we prove by refining their method. The new bound is nearly tight not only for constant degree polynomials, but actually remains so even for degree } r \text{ that is linear in } m. \]

We get a similar improvement for their bound on the list-size for list decoding of RM codes.

Summarizing, we study some very basic coding-theoretic questions regarding low-degree polynomials over \( \text{GF}(2) \). We stress two central aspects which remain elusive. First, while proving the first results about parameters of RM codes which achieve capacity, the possibly most important range, when error rate is constant, seems completely beyond the reach of our techniques. Second, while our bounds for erasures immediately entails a (trivial) efficient algorithm to actually locate them, there are no known efficient algorithms for correcting random errors in the regimes we prove it is information theoretically possible. We hope that this paper will inspire further work on the subject, and we provide concrete open questions it suggests. We now turn to give more details on the problems, results and past related work.

1.2 Notation and terminology

Before presenting our results we need to introduce some notations and parameters.

- For nonnegative integers \( r \leq m \), \( R_m(r) \) denotes the Reed-Muller code whose codewords are the evaluation vectors of all multivariate polynomials of degree at most \( r \) on \( m \) Boolean variables. The maximal degree \( r \) is sometimes called the order of the code. The blocklength of the code is \( n = 2^m \), the dimension \( k = k(m, r) \) is \( \sum_{i=0}^{r} \binom{m}{i} \), and the distance \( d = d(m, r) = 2^m - r \). The code rate is given by \( R = k(m, r)/n \).

- We use \( E(m, r) \) to denote the “evaluation matrix” of parameters \( m, r \), whose rows are indexed by all monomials of degree \( \leq r \) on \( m \) Boolean variables, and whose columns are indexed by all vectors in \( \mathbb{F}_2^m \). For \( u \in \mathbb{F}_2^m \), we denote by \( u^\top \) the column of \( E(m, r) \) indexed by \( u \), which is a \( k \)-dimensional vector, and for a subset of columns \( U \subseteq \mathbb{F}_2^m \) we denote by \( U^\top \) the corresponding submatrix of \( E(m, r) \).

- A generator matrix for \( R_m(r) \) is given by \( G(m, r) = E(m, r) \), and a parity-check matrix for \( R(m, r) \) is given by \( H(m, r) = E(m, m - r - 1) \).

- We associate with a subset \( U \subseteq \mathbb{F}_2^m \) its characteristic vector \( 1_U \in \{ 0, 1 \}^m \). We often think of the vector \( 1_U \) as denoting either an erasure pattern or an error pattern.

Finally, we use the following standard notations. \([n] = \{ 1, \ldots, n \}\). The Hamming weight of \( x \in \mathbb{F}_2^n \) is denoted \( w(x) = |\{ i \in [n] : x_i \neq 0 \}| \) and the relative weight is \( wt(x) = w(x)/n \). We use \( B(n, s) = \{ x \in \mathbb{F}_2^n : w(x) \leq s \} \) and \( \partial B(n, s) = \{ x \in \mathbb{F}_2^n : w(x) = s \} \). We use \( \binom{n}{s} \) to denote the set of subsets of \( [n] \) of cardinality \( s \). Hence, for \( S \in \binom{[n]}{s}, 1_S \in \partial B(n, s) \).

For a vector of dimension \( n \) and subset \( S \) of \( n \), we use \( x[S] \) to denote the components of \( x \) indexed by \( S \), and if \( X \) is matrix with \( n \) columns, we use \( X[S] \) to denote the subset of columns indexed by \( S \). In particular, \( E(m, r)[U] = U' \).

\[ \text{When we need to be more explicit, for an } a \times b \text{ matrix } A \text{ and } \]
I ⊆ [σ], we denote with $A_I$, the matrix obtained by keeping only those rows indexed by $I$, and denote similarly $A_{i,j}$ for $J ⊆ [b]$.

### Channels, capacity and capacity-achieving codes

We next describe the channels that we will be working with. Throughout $p$ will denote the corruption probability per coordinate. The Binary Erasure Channel (BEC) with parameter $p$ acts on vectors $v ∈ \{0,1\}^n$, by changing every coordinate to “?” with probability $p$. That is, after a message $v$ is transmitted in the BEC the received message $\hat{v}$ satisfies that for every coordinate $i$ either $\hat{v}_i = v_i$ or $\hat{v}_i = “?”$ and $\Pr[\hat{v}_i = “?”] = p$. The Binary Symmetric Channel (BSC) with parameter $p$ flips the value of each coordinate with probability $p$. That is, after a message $v$ is transmitted in the BSC the received message $\hat{v}$ satisfies $\Pr[\hat{v}_i \neq v_i] = p$.

In fact, we will use a small variation on these channels; for corruption probability $p$ we will fix the number of erasures/errors to $s = pn$. We note that by the Chernoff-Hoeffding bound (see e.g., [3]), the probability that more than $pn + \omega(\sqrt{pn})$ erasures/errors occur for independent Bernoulli choices is $o(1)$, and so we can restrict our attention to talking about a fixed number of erasures/errors. Thus, when we discuss $s$ corruptions, we will take the corruption probability to be $p = s/n$. We refer to Section 2 of the full version for the details.

We now define the notions of “capacity-achieving” for the channels above. We consider $RM(m,r)$ where $r = r(m)$ typically depends on $m$. We say that $RM(m,r)$ can correct random erasures/errs, if it can correct the random erasures/errs with high probability when $n$ tends to infinity. The goal is to recover from the largest amount of erasures/errs that is information-theoretically achievable. We note that while recovering from erasures, whenever possible, is always possible efficiently (by linear algebra), this need not be the case for recovery from errors. As we focus on the information theoretic limits, we allow maximum-likelihood (ML) decoding rule. Obtaining an efficient algorithm is a major open problem. Note that ML minimizes the error probability for equiprobable messages, hence if ML fails to decode the codewords with high probability, no other algorithms can succeed.

Recall that the capacity of a channel is the largest possible code rate at which we can recover (whp) from corruption probability $p$. This capacity is given by $1 - p$ for BEC erasures, and by $1 - h(p)$ for BSC errors. Namely, Shannon proved that for any code of rate $R$ that allows to correct corruptions of probability $p$, then $R < 1 - p$ for the BEC and $R < 1 - h(p)$ for the BSC.

Achieving capacity means that $R$ is close to the upper bound, say within $(1 + \varepsilon)$ factor of the optimal bounds above. For fixed corruption probabilities $p$ and rates $R$ in $(0,1)$ this is easy to define (previous paragraph). However as we deal with very low or very high rates above, defining this needs a bit more care, and is described in the table below. A code of rate $R$ is $\varepsilon$-close to achieve capacity if it can correct from a corruption probability $p$ that satisfies the bounds below\footnote{Note that for $R \to 0$, in the BEC we have $p \to 1$, while for the BSC we have $p \to \frac{1}{2}$. Also, we have stated the bounds thinking of $R$ fixed and putting a requirement on $p$. One can equivalently fix $p$ and require the code to correct a corruption probability $p$ for a rate $R$ that satisfies the bounds in the table.}.

### 1.3 Our results

We now state all our results, with approximate parameters, as the exact statements are somewhat technical. Next to each result we mention the corresponding exact statement in the full version of the paper [2]. We divide this section to results on decoding from random erasures, then on weight distribution and list decoding, and finally decoding random errors. In brief, we investigate four cases: two regimes for the code rates (high and low rates) and two models (BEC and BSC). Besides for the BSC at high-rate, we obtain a capacity-achieving result for all other three cases. For the low-rate regimes, we obtain results for values of $r$ up to the order of $m$.

#### 1.3.1 Random erasures - the BEC channel

As mentioned earlier, some of the questions we study concerning properties of Reed-Muller codes can be captured by the following basic algebraic-geometric questions about evaluation vectors of low-degree monomials, namely, submatrices of $E(m,r)$. The parameter $s \in [n]$ is drawn uniformly at random and denotes the size of the set $U \subseteq \mathbb{F}_2^n$:

1. What is the largest $s$ for which the submatrix $U^t$ has full column-rank with high probability?

2. What is the smallest $s$ for which the submatrix $U^t$ has full row-rank with high probability?

More generally, we will be interested in characterizing sets $U$ for which these properties hold. We note that for achieving capacity, $s$ should be as close as possible to $\binom{n}{r}$ (from below for the first question and from above for the second question). In other words, the matrix $U^t$ should be as close to square as possible. Note that this would be achieved for the case where $E(m,r)$ is replaced by a random uniform matrix, so our goal is to show that $E(m,r)$ behaves roughly like a random matrix with respect to these questions.

We obtain our decoding results for the BEC by providing answers to these questions for certain ranges of parameters. Our first theorem concerns Reed-Muller codes of low degree.

**Theorem 1** (Theorem 4.17 [2]). Let $r = o(m)$. Then, if we pick uniformly at random a set $U$ of $(1 + o(1)) \cdot \binom{n}{r}$ columns of $E(m,r)$, then with probability $1 - o(1)$ the rows of this submatrix are linearly independent, i.e., $U^t$ has full row-rank.

As an immediate corollary we get that Reed-Muller codes of sub-linear degree achieve capacity for the BEC.

**Theorem 2** (Corollary 5.1 [2]). $RM(m,r)$ achieves capacity for the BEC, for $r = o(m)$, . More precisely, for every $\delta > 0$ and $\eta = O(1/\log(1/\delta))$ the following holds: For every $r \leq \eta m$, $RM(m,r)$ is $\delta$-close to capacity for the BEC.
We obtain similar results in a broader range of parameters when the code is of high degree rather than low degree (i.e., the code has high rate rather than low rate).

Theorem 3 (Theorem 4.5 [2]). If we pick uniformly at random a set $U$ of $(1 - o(1)) \cdot \binom{m}{\ell}$ columns of $E(m, r)$, for $r = O(\sqrt{m/\log m})$, then with probability $1 - o(1)$ they are linearly independent, i.e., the submatrix $U^\ell$ has full column rank.

Due to the duality between linear independent set of columns in $E(m, r)$ and spanning sets in the generating matrix of $RM(m, m - r - 1)$ (see Lemma 4.3 [2]) we get as corollary that Reed-Muller codes with the appropriate parameters achieve capacity for the BEC.

Theorem 4 (Corollary 5.2 [2]). For $m - r = O(\sqrt{m/\log m})$, $RM(m, r)$ is capacity-achieving on the BEC.

1.3.2 Weight distribution and list decoding

Before moving to our results on random errors, we take a detour to discuss our results on the weight distribution of Reed-Muller codes as well as their list decoding properties. These are naturally important by themselves, and, furthermore, tight weight distribution bounds turn out to be crucial for achieving capacity for the BEC in Theorem 1 above, as well as for achieving capacity for the BSC in Theorem 7 below. Our bound extends an important recent result of Kaufman, Lovett and Porat on the weight-distribution of Reed-Muller codes [27], using a simple variant of their technique. Kaufman et al. gave a bound that was tight for $r = O(1)$, but degrades as $r$ grows. Our improvement extends this result to degrees $r = O(m)$. Denote with $W_{m, r}(\alpha)$ the number of codewords of $RM(m, r)$ that have at most $\alpha$ fraction of nonzero coordinates.

Theorem 5 (Theorem 3.3 [2]). Let $1 \leq \ell \leq r - 1 < m/4$ and $0 < \varepsilon \leq 1/2$. Then,

$$W_{m, r}((1 - \varepsilon)2^{-\ell}) \leq (1/\varepsilon)^{O(\ell^4 \binom{m - \ell}{r - \ell})}.$$ 

As in the paper of [27], almost the exact same proof as our proof of Theorem 5 yields a bound for list-decoding of Reed-Muller codes, for which we get similar improvements. Following [27] we denote:

$$L_{m, r}(\alpha) = \max_{g : \{0, 1\}^m \to \{0, 1\}} \{ f \in RM(m, r) \mid \text{wt}(f - g) \leq \alpha \}. $$

That is, $L_{m, r}(\alpha)$ denotes the maximal number of code words of $RM(m, r)$ in a hamming ball of radius $\alpha2^m$. The bound concerns $\alpha$ of the form $(1 - \varepsilon)2^{\ell}$ for $1 \leq \ell \leq r - 1$, and our main contribution is making the first factor in the exponent depend on $\ell$ (rather than on $r$ in [27]).

Theorem 6. Let $1 \leq \ell \leq r - 1$ and $0 < \varepsilon \leq 1/2$. Then, if $r \leq m/4$ then

$$L_{m, r}((1 - \varepsilon)2^{-\ell}) \leq (1/\varepsilon)^{O(\ell^4 \binom{m - \ell}{r - \ell})}.$$ 

1.3.3 Random errors - the BSC channel

We now return to discuss decoding from random errors. Our next result shows that Reed-Muller codes achieve capacity also for the case of random errors at the low rate regime. The proof of this result relies on Theorem 5.

Theorem 7 (Theorem 6.1 [2]). For $r = o(m)$, $RM(m, r)$ achieves capacity for the BSC. More precisely, for every $\delta > 0$ and $\eta = O(1/\log(1/\delta))$ the following holds: For every $r \leq \eta m$, $RM(m, r)$ is $\delta$-close to capacity for the BSC.

To obtain results about the behavior of high-rate Reed-Muller codes with respect to random errors we use a novel connection between robustness to errors and robustness to erasures in related Reed-Muller codes.

Theorem 8 (Theorem 6.14 [2]). If a set of columns $U$ are linearly independent in $E(m, r)$ (namely, $RM(m, m - r - 1)$) can correct the erasure pattern $1_U$), then the error pattern $1_U$ can be corrected (i.e., it is uniquely decodable) in $RM(m, m - (2r + 2))$.

Using Theorem 3 this gives a new result on correcting random errors in Reed-Muller codes.

Theorem 9 (Theorem 6.2 [2]). $RM(m, m - (2r + 2))$ can correct a random error pattern of weight $(1 - o(1)) \cdot \binom{m}{r}$ with probability larger than $1 - o(1)$, when $r = O(\sqrt{m/\log m})$.

While this result falls short of showing that Reed-Muller codes achieve capacity for the BSC in this parameter range, it does show that they can cope with many more errors than suggested by their minimum distance. Recall that the minimum distance of $R(m, m - (2r + 2))$ is $2^{2r + 2}$. Achieving capacity for this code means that it should be able to correct roughly $\binom{m}{r}$ random errors. Instead we show that it can handle roughly $\binom{m}{r}$ random errors, which is approximately the square root of the number of errors at capacity.

The proof of Theorem 8 reveals a more general phenomenon, that of reducing error correction to erasure correction. We prove that for any linear code $C$, of very high rate, there is another linear code $C'$ of related high rate, so that if $C$ can correct the erasure pattern $1_U$ then $C'$ can correct the error pattern $1_U$. Furthermore $C'$ is very simply defined from $C$. The decline in quality of $C'$ relative to $C$ is best explained in terms of the co-dimension (i.e., the number of linear constraints on the code, or equivalently the number of rows of its parity-check matrix). We prove that the co-dimension of $C'$ is roughly the cube of the co-dimension of $C$. We now state this general theorem.

For a matrix $H$ we denote by $H^3$ the corresponding matrix that contains the evaluations of all columns of $H$ by all degree $\leq r$ monomials (in an analogous way to the definition of $U^\ell$ from $U$).

Theorem 10 (Theorem 6.18 [2]). If a set of columns $U$ is linearly independent in a parity check matrix $H$, then the code that has $H^3$ as a parity check matrix can correct the error pattern $1_U$.

Note that applying this result as is to $E(m, r)$ would give a weaker statement than Theorem 8, in which $E(m, 2r + 1)$ would be replaced by $E(m, 3r)$. We conclude by showing that this result is tight, i.e., replacing 3 by 2 in the theorem above fails, even for RM codes.

Theorem 11 (Section 6.25 [2]). There are subsets of columns $U$ that are linearly independent in $E(m, 1)$, but such that the patterns $1_U$ are not uniquely decodable in $E(m, 2)$.
1.4 Proof techniques

Although the statements of Theorems 3 and 1 sound very similar, their proofs are very different. We first explain the ideas behind the proofs of these two theorems and then give details for the proofs of Theorems 5, 7, 8 and 10.

Proof of Theorem 3.

The proof of Theorem 3 relies on estimating the size of varieties (sets of common zeros) of linear subspaces of degree \( r \) polynomials. Here is a high level sketch.

Recall that we have to show that if we pick a random set of points \( U \subset \mathbb{F}_q^m \), of size \( (1-o(1)) \cdot \left( \begin{array}{c} m \cr r \end{array} \right) \), and with each point associate its degree-\( r \) evaluation vector, then with high probability these vectors are linearly independent. While proving this is simple when considered over large fields, it is quite non-trivial over very small fields. We are able to prove that this property holds for degrees \( r \) up to (roughly) \( \sqrt{m}/\log m \). It is a very interesting question to extend this to larger degrees as well.

To prove that a random set \( U \) of appropriate size generates rise to linearly independent evaluation vectors we consider the question of what it takes for a new point to generate an evaluation vector that is linearly independent of all previously generated vectors. As we prove, this boils down to understanding what is the probability that a random point is a common zero of all degree \( r \) polynomials, in a certain linear space of polynomials defined by the previously picked points. If this set of common zeros is small, then the success probability (i.e., the probability that a new point will yield an independent evaluation vector) is high, and we can iterate this argument.

To bound the number of common zeros we yet again move to a dual question. Notice that if a set of \( K \) linearly independent polynomials of degree \( r \) vanishes on a set of points \( V \), then there are at most \( \left( \begin{array}{c} m \cr r \end{array} \right) - |K| \) linearly independent degree \( r \) polynomials that are defined over \( V \). In view of this, the way to prove that a given set of polynomials does not have too many common zeros is to show that any large set of points (in our case, the set of common zeros) has many linearly independent degree \( r \) polynomials that are defined over it. We give two different proofs of this fact. The first uses a hashing argument; if \( V \) is large then after some linear transformation it supports many different degree \( r \) monomials. The second relies on a somewhat tighter bound that was obtained by Wei [43], who studied the generalized Hamming weight of Reed-Muller codes. While Wei’s result gives slightly tighter bounds compared to the hashing argument, we find the latter argument more transparent.

Proof of Theorem 1.

To prove Theorem 1 we first observe that a set of columns \( U \) (in \( E(m, r) \)) spans the entire row-space if and only if there is no linear combination of the rows of \( E(m, r) \) that is supported on the complementary set \( U^c = \mathbb{F}_q^m \setminus U \). As linear combinations of rows correspond to “truth-tables” of degree \( r \) polynomials, this boils down to proving that, with high probability, no nonzero degree \( r \) polynomial vanishes on all points in \( U \). For each such polynomial, if we know its weight (the number of nonzero values it takes), this is a simple calculation, and the hope is to use a union bound over all polynomials. To this end, we can partition the codewords to dyadic intervals according to their weights, carry out this calculation and union bound the codewords in each interval and then combine the results. For this plan to work we need a good estimate of the number of codewords in each dyadic interval, which is what Theorem 5 gives.

Proof of Theorem 5.

As mentioned earlier, this theorem improves upon a beautiful result of Kaufman, Lovett and Porat [27]. Our proof is closely related to their proof. Roughly, what they show is that any small weight codeword, i.e., a degree \( r \) polynomial with very few non-zero values, can be well approximated by a “few” partial derivatives. In particular, there is a function that when applied to a few lower degree polynomials, agrees with the original polynomial on most of the inputs. Here “few” depends on the degree \( r \), the weight and (crucially for us) the quality of the approximation. Kaufman et al. then pick an approximation quality parameter that guarantees that the approximating function can be close to at most one polynomial of degree \( r \). Then, counting the number of possible approximating functions they obtain their bound. The cost is that such a small approximation parameter blows the number of “few” derivatives that are required. We diverge from their proof in that we choose a much larger approximation quality parameter, but rather allow each approximating function to be close to many degree \( r \) polynomials. The point is that, by the triangle inequality, all these polynomials are close to each other, and so subtracting one of them from any other still yield polynomials of very small weight. Thus, we can use induction on weight to bound their number, obtaining a better bound on the number of polynomials of a given weight.

Proof of Theorem 7.

Here we use a union bound argument on the error probability and show that the bound is still tight enough to ensure the capacity-achieving property for RM codes in the considered regime. To prove that the code can, w.h.p., uniquely decode an error pattern \( 1_U \) of weight \( w \), we basically wish to show that for no other error pattern \( 1_V \), of weight \( w \), the vector \( 1_U \oplus 1_V \) is a code word (as then both error patterns will have the same syndrome). Stated differently, we want to count how many different ways are there to represent a codeword as a sum of two vectors of weight at most \( w \). This counting depends very much on the weight of the codeword that we wish to split. In random linear codes weights are very concentrated around \( n/2 \), which makes the union bound easy. Reed-Muller codes however have many more codewords of smaller weights, and the argument depends precisely on how many. Once again Theorem 5 comes to the rescue and enables us to make this delicate calculation for each (relevant) dyadic interval of weights. Here too our improvement of [27] is essential.

Proofs of Theorems 8, 9 and 10.

Consider an erasure pattern \( 1_U \) such that the corresponding set of degree-\( r \) evaluation vectors, \( U^c \), is linearly independent. I.e., the columns indexed by \( U \) in \( E(m, r) \) are linearly independent. We would like to prove that \( 1_U \) is uniquely decodable from its syndrome under \( H(m, m - 2r - 2) = E(m, 2r + 1) \). We actually prove that if \( 1_V \) is another erasure pattern, which has the same syndrome under \( H(m, m - 2r - 2) \), then \( U = V \). The proof may be viewed as a reconstruction (albeit inefficient) of the set \( U \) from its
syndrome. Here is a high level description of our argument that different (linearly independent) sets of erasure patterns
give rise to different syndromes.

We first prove this property for the case \( r = 1 \) (details below). This immediately implies Theorem 10 as every parity
check matrix of any linear code is a submatrix of \( E(m, 1) \)
for some \( m \). This is a general reduction from the problem
of recovering from errors to that of recovering from erasures
(in a related code). As a special case, it also implies that
for any \( r, \) \( H(m, m - 3r - 1) \) uniquely decodes any error pattern
\( V \) such that the columns indexed by elements of \( U \) in
\( E(m, r) = H(m, m - r - 1) \) are linearly independent. We
then slightly refine the argument for larger degree \( r \) to re-
place \( H(m, m - 3r - 1) \) above by \( H(m, m - 2r - 2) \), which
gives Theorem 8.

For the case \( r = 1 \), the proof divides to two logical steps.
In the first part we prove that the columns of \( V \) must span
the same space as the columns of \( U \). This requires only the
submatrix \( E(m, 2) \), i.e., looking at pairs of coordinates in
each point (degree 2 monomials). In the second part we use
this property to actually identify each vector of \( U \) inside \( V \).
This already requires looking at the full matrix \( E(m, 3) \), i.e.,
at triples of coordinates.

It is interesting that going to triples of coordinates is es-
ential for \( r = 1 \) (and so this result are tight). We prove that
even if the columns of \( U \) are linearly independent, then
there can be a different set \( V \) that has the same syndrome
in \( E(m, 2) \). We do not know what is the right bound for
general \( r \).

1.5 Related literature

Recovery from random corruptions

Besides the conjectures mentioned in the introduction
that RM codes achieve capacity, results fall short of that
for all but very spacial cases. We are not familiar of works
correcting random erasures. Several papers have considered
the quality of RM codes for correcting random errors when
using specific algorithms, focusing mainly on efficient algo-
rithms. In [28], the majority logic algorithm [36] is shown
to succeed in recovering all but a vanishing fraction of er-
ror patterns of weight up to \( (4/\ln(4))r \), where \( d = 2^{m-r} \).
This immediately implies Theorem 10 as every parity

Theorem 10: For all but a vanishing fraction of erasure
patterns of weight up to \( (4/\ln(4))r \), the majority logic
algorithm decodes the \( m \) even-weight errors.

We prove this result using Theorem 10 as a black box.

For the special case of \( r = 1,2 \) (i.e., the generator matrix
has only vectors of weights \( n/2 \) and \( n/4 \)), [24] shows that
RM codes are capacity-achieving. For \( r \geq 3 \), the problem is
left open.

Weight enumeration

The weight enumerator (how many codewords are of any
given weight) of \( RM(m, 2) \) was characterized in [41]. For
\( RM(m, 3) \), a complete characterization of the weight enu-
merator is still missing. The number of codewords of mini-
mal weight is known for any \( r \), and corresponds to the num-
ber of \( (m-r) \)-flats in the affine geometry \( AG(m, 2) \). [31].
In [25], the weight enumerator of RM codes is characterized
for codewords of weight up to twice the minimum distance,
later improved to 2.5 the minimum distance in [26].

For long, [26] remained the largest range for which the
weight enumerator was characterized, until [27] managed to
breakthrough the 2.5 barrier and obtained bounds for all
distances in the regime of small \( r \).

2. FUTURE DIRECTIONS AND OPEN

PROBLEMS

We believe that our work renewes hope for progress on
some classical questions, and suggests some new concrete
directions and open problems.

The most obvious of all is the question of whether Reed-
Muller codes achieve capacity for all ranges of parameters,
either for random erasures or for random errors. We only
handle here the extreme cases of very high or very low rates,
whereas most interest is traditionally focused on constant
rate codes. We believe that the techniques for each of our
four bounds can be improved to a larger set of parameters
(see below), but feel that they fall short of reaching constant
rate, and possibly new techniques are needed.

One way to improve our bounds in both Theorem 1 (low-
rate BEC) and Theorem 7 (low rate BSC) is through tighter
bounds on the weight enumeration of Reed-Muller codes, as
well as tighter bounds on the probability of error for The-
orem 7. We believe that in Theorem 5 one can eliminate
the factor \( \ell^r \) in the exponent, resulting in a bound that is a
fixed polynomial (independent of \( m, r, \ell \)) of the lower bound
in [27]. While such a tight result would not get us (in either
Theorem 1 and 7) to the constant rate regime, this question
of weight enumeration is of course basic in its own right.
Moreover, both in [27] and our paper, it also implies simi-
lar bounds for list-decoding, which is another basic question.

Theorem 4 (high rate BEC) is quantitatively much weaker
than Theorems 1 and 7, in that the latter two can handle
polynomials of degree-\( r \) which is linear in \( m \), whereas
the former only reaches degrees \( r \) which are about \( \sqrt{m} \). The
bottleneck in the argument, which probably prevents it from
reaching a linear degree, is the use of the union bound. We
upper bound the probability that, when adding a subsequent
random vector \( u \) to our set \( U \), its evaluation \( u^r \) will be lin-
early independent of the evaluations of all previously chosen
points. This current proof does not use at all that previous

303
points were chosen randomly, as we don’t know how to take advantage of this.

For high-rate BSC (Theorem 9), while we are able to correct many more errors than previously known, we are not even able to achieve capacity. Here we feel that one important bottleneck is our inability to argue directly about corruption patterns (sets $U$) which are linearly dependent. Our unique decoding proof, even for $r = 1$ (on which we focus now), showing that a set $U \in \mathbb{F}_q^n$ is uniquely determined by its syndrome under evaluations by degree-3 monomials i.e., by $E(m, 3) \cdot 1_U$, is especially tailored to linearly independent sets $U$. The gap between our lower bound (i.e., that $E(m, 2) \cdot 1_U$ does not suffice) and the above upper bound (that $E(m, 3) \cdot 1_U$ suffices) is intriguing, and we believe we can find a subset of quadratically many monomials of degree at most 3 which guarantee unique decoding - such a result is information theoretically optimal; number of error patterns $U$ which are linearly independent is about $\exp(m^2)$, and thus $O(m^2)$ bits are needed in any unique encoding.

Another burning question regarding this result is its in-efficiency. While unique decoding is guaranteed, the best way we know to identify the set $U$ is brute force, requiring $\exp(m^2)$ steps for independent sets $U$ of size $m$. We feel that a good starting place is (perhaps using our uniqueness proof) which recovers $U$ in $\exp(m) = \text{poly}(n)$ steps from its evaluation on all degree-3 monomials (or even degree-10 monomials). Of course, it is quite possible that a poly($m$) algorithm exists. In particular, recursive algorithms (that exploit the recursive nature of RM codes) could be used to that effect.

Yet another research direction related to our result is of course exploring the connections between recovering from erasures and from errors. Our general reduction between the two uses tensor powers and hence loses in efficiency (which here is best captured by the co-dimension of the code, which is cubed). Is there a reduction which losess less? We do not know how to rule out a reduction that increases the co-dimension only by a constant factor. There is no reason to restrict attention to Reed-Muller codes and our tensor construction - such a result would be of use anywhere, as erasures are so much simpler to handle than errors.

Finally, we believe that a better understanding of the relation between Reed-Muller codes and Polar codes is needed, and perhaps more generally an understanding of which subspaces of polynomials generated by subsets of monomials give rise to good, efficient codes. In particular, it would also be interesting to investigate the scaling of the blocklength in terms of the gap to capacity for RM codes. It was proved recently in [21] that for polar codes, the blocklength scales polynomially with the inverse of the gap to capacity, with a precise characterization given in [23]. While this scaling does not match the optimal scaling of random codes [42], it is in contrast to the exponential scaling obtained with concatenated codes [18] (see [21] for a discussion on this). It would be interesting to investigate such finer questions for RM codes in view of the results obtained in this paper, which already provide partial information about these scalings.

3. **HIGH-RATE RM CODES FOR THE BSC**

In this section we give the main ideas behind the proof of our main result for the BSC in the high rate regime. We first give a combinatorial view of the syndrome of an error pattern under $E(m, r)$ (Section 3.1) and then study the case of $E(m, 3)$, which corresponds to the case $r = 1$ in Theorem 9 (as $E(m, 3) = H(m, m - 4)$). The case of higher degree follows similarly and is omitted from this version.

### 3.1 Parity check matrix and parity of patterns

In this section we give a combinatorial interpretation of the syndrome of an error pattern. Consider the code $RM(m, m - r - 1)$. Its parity check matrix is $H(m, m - r - 1) = E(m, r)$.

Let $U \subseteq \mathbb{F}_q^m$ be a set of size $s$. We associate with $U$ the error pattern $1_U \in \mathbb{F}_q^m$. Clearly $w(1_U) = |U| = s$. We denote with $u_j$ the $j$'th element of $U$. We shall also think of $U$ as an $m \times s$ matrix whose $j$'th column is $u_j$. We denote with $U'$ the submatrix of $E(m, r)$ whose columns are indexed by $U$. Alternatively, this is the set of all evaluation vectors of $U$’s columns. We shall use the same convention for another subset $V \subseteq \mathbb{F}_q^m$.

The following definition captures a combinatorial property that we will later show its relation to syndromes under $E(m, r)$.

**Definition 1.** For two matrices $A, B$ of same dimension $n_1 \times n_2$, we denote $A \sim B$ if any pattern of size at most $r$ in the columns of $A$ appears with the same parity in the columns of $B$. I.e., for every subset $I \subseteq [n_1]$ of size $r$ and every $z \in \mathbb{F}_q^r$ the number of columns in $A_I$ that equal $z$ is equal, modulo 2, to the number of columns in $B_I$ that equal $z$.

The next lemma shows that two error patterns $1_U$ and $1_V$ have the same syndrome under $E(m, r)$ if and only if the two matrices $U$ and $V$ satisfy $U \sim V$.

**Lemma 1 (Parity of patterns).** For two sets $U, V \subseteq \mathbb{F}_q^m$ of size $s$ it hold that $E(m, r) \cdot 1_U = E(m, r) \cdot 1_V \iff U \sim V$.

Our goal is to understand, for given values of $m$ and $r$, how many vectors $1_U$ are bad, in the sense that they admit a bad companion $1_V$ such that $E(m, r) \cdot 1_U = E(m, r) \cdot 1_V$. Thus, by Lemma 1 this is equivalent to studying pairs $U, V$ such that $U \sim r, V$.

### 3.2 The case $r = 1$

We will prove first a deterministic result: if $U$ is any set of linearly independent columns, then for any $V \neq U$, we have that $1_U \neq 3 V$. Thus, any set of errors that is supported on linearly independent coordinates (when viewed as vectors in $\mathbb{F}_q^m$) can be uniquely corrected. This immediately gives an average-case result. If we have $m - \log(m/e)$ random errors, then with probability at least $1 - \varepsilon$ their locations correspond to linearly independent $m$-bit vectors and therefore we can correct such amount of errors with high probability. Notice that this is already highly nontrivial, as $R(m, m - 4)$ has (absolute) distance 16, so in the worst case one cannot correct more than 8 worst-case errors!
Lemma 2. Let $U \subseteq \mathbb{F}_2^n$ be a set of linearly independent vectors, such that $|U| = s$. Then, for any $V \neq U$, such that $|V| \leq s$, we have that $V \not\sim U$.

In particular this means that we can correct the error pattern $1_U$ in $RM(m, m-4)$.

Proof. For simplicity we assume that the columns of $U$ are the elementary basis vectors, $e_1, \ldots, e_s$.\(^{10}\)

Let $V \subseteq \mathbb{F}_2^n$ be such that $|V| = s$ and $V \sim U$. Our task is to show that $V = U$. This will be shown in two steps. First, we will show that span$(V)$ = span$(U)$, which in particular implies that $V$ is linearly independent as well. Proving linear independence requires only that $V \not\sim U$, i.e., looking only at evaluations by degree-2 monomials. Using $V \sim U$, we will prove that they actually have the same span, and from that derive that $V = U$.

Consider the pattern (1,0) in the first two rows of $U$. That is, consider all columns of $U$ that have 1 in their first coordinate and 0 in the second. It is clear that this pattern only appears in $e_1$ and hence its parity in $U$ is 1. Thus, there must be an odd number of columns in $V$ whose first two rows equal (1,0). The main observation is that if we add up the columns then we obtain the vector $e_1$.

Claim 1. Under the conditions of the lemma, the sum of all columns in $V$ whose first two coordinates equal (1,0) is $e_1$. More generally, for $i \in [s]$, if we consider the pattern that has 1 in the $i$’th coordinate and 0 in some $j \neq i$ coordinate, then the sum of all columns in $V$ that have this pattern is equal to $e_i$.

Proof. Assume that this is not the case, i.e., that the sum is a vector $w \neq e_1$. We first note that the first two coordinates of $w$ equal (1,0). Indeed, this holds as we summed an odd number of vectors that has these values. Hence, there must exist a coordinate $i > 2$ such that $w_i = 1$. Thus, the number of vectors in $V$ with the pattern (1,0,1) in rows (1,2,i) is odd. But this is not the case in $U$, contradicting $V \sim U$.

The proof of the general case is similar. □

As an immediate corollary we obtain that the columns of $V$ are linearly independent.

Claim 2. Under the conditions of the lemma, it must be that the columns in $V$ are linearly independent and $|V| = s$.

Proof. Claim 1 implies that $e_i$ is in the columns span of $V$ for all $i \in [s]$. As $|V| \leq s$ the claim follows. □

We now use the fact that $U$ and $V$ have the same span to conclude that $U = V$. Denote $J_{10} \subseteq [s]$ the indices of columns in $V$ that have (1,0) as their first two coordinates. By Claim 1 we have that $\sum_{i \in J_{10}} v_i = e_1$. Next, consider the pattern (1,*,0), i.e., the pattern that has 1 in the first row and 0 in the third row. Denote the corresponding set of column indices with $J_{1*0}$. Again, Claim 1 implies that $\sum_{i \in J_{1*0}} v_i = e_1$. However, since the columns in $V$ are linearly independent, there is only one way to represent $e_1$ as a linear combination of the columns of $V$ and therefore it must be the case that $J_{10} = J_{1*0}$.

Continuing in this fashion we get that $J_{10} = J_{1*0} = J_{1**0} = \ldots = J_{1**n}$, where the last set corresponds to the columns that have 1 in the first coordinate and 0 in the last coordinate. As the size of $J_{10}$ is odd we know that it is not empty. In particular, all the vectors in $J_{10}$ must satisfy that they have 1 in the first coordinate and 0 in the remaining coordinates. Indeed each such 0 can be justified by one of those $J$ sets. In particular $e_1$ is a column in $V$. Repeating this process for all $e_i$, $i \in [s]$, we get that all these $e_i$’s are columns in $V$. Since $V$ has exactly $s$ columns it must have the same set of columns as $U$. In particular, $V = U$. This completes the proof of Lemma 2. □

Lemma 2 shows that if the columns of $U$ are linearly independent then $E(m, 3) \neq E(m, 3)_{V}$ for any other $V$ of the same size. As randomly picking $m - \log(m/\varepsilon)$ vectors in $\mathbb{F}_2^n$ we are likely to get linearly independent vectors we obtain our main result for the case $r = 1$.

4. ACKNOWLEDGEMENTS

We thank Venkatesan Guruswami for bringing [43] to our attention and Iwan Duursma and Ilya Dumer for their comments. The second author would like to thank the organizers of Dagstuhl meeting “Algebra in Computational Complexity,” where he discussed the results with Venkatesan Guruswami. This research of the first author was partially supported by NSF grant CIF-1706648, the second author by the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575, and from the Israel Science Foundation (grant number 339/10), and the third author by NSF grant CCF-1412958.

5. REFERENCES

[1] E. Abbe. Randomness and dependencies extraction via polarization. In ITA, pages 1–7, 2011. Available at arXiv:1102.1247.
[2] E. Abbe, A. Shpilka, and A. Wigderson. Reed-Muller codes for random erasures and errors. CoRR, abs/1411.4590, 2014. To appear in STOC 2015.
[3] N. Alon and J. Spencer. The Probabilistic Method. John Wiley, 1992.
[4] E. Arikan. A performance comparison of polar codes and Reed-Muller codes. Communications Letters, IEEE, 12(6):447–449, June 2008.
[5] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. Information Theory, IEEE Transactions on, 55(7):3051–3073, 2009.
[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. Journal of the ACM (JACM), 45(3):501–555, 1998.
[7] A. Ashikhmin and S. Litsyn. Simple MAP decoding of first-order Reed-Muller and Hamming codes. Information Theory, IEEE Transactions on, 50(8):1812–1818, Aug 2004.
[8] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. In Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on, pages 16–25. IEEE, 1990.
