Algebraic Properties of Tensor Product Matrices, with Applications to Coding

Yuval Wigderson

Advised by Emmanuel Abbe

Thesis submitted to the Department of Mathematics, Princeton University, in partial fulfillment of the requirements for the degree of Bachelor of Arts

May 2, 2016

This thesis represents my own work in accordance with University regulations.

Yuval Wigderson

Dedicated to my grandmother, Miryam Rothenstein

Acknowledgments

First and foremost, I'd like to thank Emmanuel Abbe, without whom no part of this thesis would exist. His support and guidance throughout the past three years have been invaluable, and I couldn't have done it without him.

I'd also like to thank my family, who have put up with my nonsense my whole life, my friends, for making my experience at Princeton unforgettable, and my professors, for teaching me.

This project was supported by Princeton's PACM Director's Fellowship, NSF grant CIF-1706648, and the NSF Center for the Science of Information's REU program.

"I feel I understand
Existence, or at least a minute part
Of my existence, only through my art,
In terms of combinatorial delight;"

Vladimir Nabokov, *Pale Fire*

Contents

1 Introduction

Coding theory is, fundamentally, the study of communication in the presence of noise, or equivalently the study of error-correction. In the context of coding, we have some message that we wish to transmit through a medium that can corrupt it, and we want to introduce redundancy into what we transmit so that the original message can be recovered from its noisy version. Any method for doing this is called a *coding scheme,* and the messages that we send are called *codewords.* The study of error-correcting codes was begun by Claude Shannon

in his seminal 1948 paper, [15]. In this paper, Shannon formally defined the central problem of error-correction and proved that protection against noise was indeed possible. The ideas he introduced have found application in almost every aspect of digital technology, and are invaluable whenever communication or storage is necessary.

By far the simplest and most thoroughly-studied codes are *linear codes*; in these codes, the alphabet in which our messages are written is a field, and the codewords form a vector space over this field. Linear codes are enormously useful for a variety of reasons, both practical and theoretical, and this is why they feature so prominently in coding theory research; nevertheless, they are often studied using more general coding-theoretic techniques that do not explicitly use their linear structure. In this thesis, the goal is to study certain classes of linear codes from a purely linear-algebraic point of view, and to demonstrate that such an analysis can provide interesting insights into the nature of these codes.

More formally, the setup is as follows:

Definition 1.1. A *code* with parameters $[n, k]$ over an alphabet $\mathscr{X}$, where $\mathscr{X}$ is any finite set, is subset $C \subseteq \mathscr{X}^n$ with $|C| = |\mathscr{X}|^k$. The elements of C are called *codewords*. n is called the *blocklength* of the code, and k is called the *message length* or *dimension* of the code. The quantity k/n is called the *rate* of the code (denoted $r(C)$), and measures how much redundancy we're introducing; the smaller the rate, the more redundancy we have. A code is called *linear* if $\mathscr{X}$ is a field and C is a k-dimensional vector subspace of $\mathscr{X}^n$. In this case, we will often write $\mathbb{F}$ for $\mathscr{X}$.

We will be primarily interested in codes that can correct erasures, which are corruptions in which transmitted message has some of its coordinates erased. More formally,

Definition 1.2. The *memoryless erasure channel* (MEC) with parameter $p \in [0, 1]$, denoted $MEC(p)$, is the channel that takes an incoming message $x \in \mathscr{X}^n$ and returns a message $y \in (\mathscr{X} \cup \{?\})^n$, where each coordinate of x is replaced with a ? with probability p and is left unchanged with probability $1 - p$, and this happens independently in each coordinate. In the special case where $\mathscr{X}$ is the binary field $\mathbb{F}_2$, we will speak of the *binary erasure channel* $BEC(p)$.

Since we are dealing with probabilistic erasures, our notion of successful transmission must also be probabilistic:

Definition 1.3. Fix a code C and a parameter $p \in [0,1]$. Pick a codeword $x \in C$ uniformly at random, and let $y \in (\mathcal{X} \cup \{?\})^n$ be the output of x under the $MEC(p)$. The *probability of error* of decoding C under the $MEC(p)$ is defined as

$$P_e(C,p) := \mathop{\mathbb{P}}_{x,y}(\exists x' \in C, x' \neq x, \text{ such that } x' \text{ can output } y \text{ under the } MEC(p))$$

This is precisely the probability that when we transmit a random message from C, we end up with an output that cannot be uniquely decoded.

Given a family of codes $\mathscr{C} = \{C_n\}_{n\geq 1}$, we say that $\mathscr{C}$ can successfully transmit on the $MEC(p)$ if

$$\lim_{n\to\infty} P_e(C_n,p) = 0$$

A famous theorem of Shannon [15] precisely tells us when we can hope to successfully transmit over the $MEC(p)$:

Theorem 1.1 (Shannon [15])**.** *For any $R < 1-p$, there exists a family of codes $\mathscr{C} = \{C_n\}_{n\geq 1}$ with rates $r(C_n) \geq R$ such that $\mathscr{C}$ can successfully transmit on the $MEC(p)$. On the other hand, for any $R' > 1-p$ and any family of codes $\mathscr{C}' = \{C'_n\}_{n\geq 1}$ with $r(C'_n) \geq R'$, we have that $P_e(C'_n,p)$ is bounded away from zero for all n.*

The number $1-p$ is called the Shannon capacity of the $MEC(p)$.

In other words, at rates lower the Shannon capacity, we can find coding schemes that can protect against probability-p erasures, but at rates greater than the Shannon capacity, any coding scheme is doomed to fail. This motivates the following definition:

Definition 1.4. A family of codes $\mathscr{C} = \{C_n\}_{n\geq 1}$ is called *capacity-achieving* on the $MEC(p)$ if $\mathscr{C}$ can successfully transmit on the $MEC(p)$ and

$$\lim_{n\to\infty} r(C_n) = 1-p$$

Shannon's proof that capacity-achieving codes exist was probabilistic, and therefore non-constructive; he demonstrated that such families exist, but could not say how to find them, nor how to make them efficiently computable (and thus practically useful). Therefore, much of the project of coding theory in the past decades has been to try to find such capacity-achieving codes.

The rest of the thesis is divided into three major sections. The first introduces the various important linear-algebraic concepts that will be used, the second discusses their application to polar codes, and the third discusses their application to Reed-Muller codes.

1.1 Notations

We use the following (mostly standard) notations throughout.

- For a positive integer n, $[n]$ denotes the set $\{1, \ldots, n\}$.
- Given a matrix A with n columns and a set $P \subseteq [n]$, $A[P]$ denotes the submatrix of A obtained by only keeping those columns indexed by P.
- Given an $m \times n$ matrix A and a parameter $p \in [0,1]$, $A[p]$ denotes the random submatrix of A obtained by selecting columns of A independently with probability p; more formally, $A[p]$ is just $A[P]$ where $P \subseteq [n]$ is sampled according to the product Bernoulli distribution $Ber(p)^n$. Similarly, if $\underline{p} = (p_1, \ldots, p_n)$ is a vector of probabilities, then $A[\underline{p}]$ is the random matrix obtained by keeping the jth column of A with probability p_j.
- Given a matrix A with m rows and an index $i \in [m]$, A_i denotes the ith row of A; similarly, $A_{\sim i}$ denotes the submatrix of A obtained by removing the ith row. Finally, $A^{(i)}$ denotes the first i rows of A.
- $\mathbb{P}(B)$ denotes the probability of an event B, and $\mathbb{E}(X)$ denotes the expectation of a random variable X.
- Given a sequence of events $\{B_n\}_{n \geq 1}$, we say that the sequence happens *with high probability* (whp) if

$$\lim_{n \to \infty} \mathbb{P}(B_n) = 1$$

- All logarithms are in base 2.

2 Algebraic Measures of Independence

2.1 Definitions

In this section, we will define and begin to study four algebraic notions of independence associated to matrices, whose properties and relationships to one another will be used later to prove various coding-theoretic results. In everything that follows, A is an $m \times n$ matrix over some field $\mathbb{F}$, $p \in [0,1]$ is some parameter, $\underline{p} = (p_1, \ldots, p_n) \in [0,1]^n$ is a vector of probabilities, $i \in [m]$ is

a row index, and $j \in [n]$ is a column index. Since we are working over a fixed field $\mathbb{F}$, all notions of linear dependence and independence are considered over $\mathbb{F}$.

Definition 2.1 (Originally defined in [1])**.** The *conditional rank* (COR) associated to A, p, and row i is

$$\rho_i(A,p) := \mathbb{P}(\text{row } i \text{ of } A[p] \text{ is linearly independent of the previous } i-1 \text{ rows})$$

Definition 2.2. The *doubly-conditional rank* (DOR) associated to A, p, and row i is

$$\Psi_i(A,p) := \mathbb{P}(\text{row } i \text{ of } A[p] \text{ is linearly independent of the other } m-1 \text{ rows})$$

Note that the definition of DOR is almost identical to that of COR, with the only difference being that COR only considers the previous rows of the matrix, while DOR considers all other rows.

Definition 2.3 (Originally defined in [5])**.** The *vector EXIT function* associated to A, $\underline{p}$, and column j is

$$\underline{h}_j(A,\underline{p}) := \mathbb{P}(\text{column } j \text{ of } A \text{ is linearly dependent on the columns in } A[\underline{p}_{\sim j}])$$

where $A[\underline{p}_{\sim j}]$ means that we discard the jth column of A and select each other column with the probability indicated by $\underline{p}$.

Note that the vector EXIT function is one whose argument is a vector and whose output is a scalar. We will also be interested in the *scalar EXIT function*

$$h_j(A,p) := \underline{h}_j(A,(p,\ldots,p))$$

gotten by substituting a uniform probability vector $\underline{p} = (p,\ldots,p)$ into $\underline{h}_j$.

Definition 2.4 (See, e.g., [12])**.** The *probability of bit-error* associated to A, p, and column j is

$$P_{e,j}(A,p) := p \cdot h_j(A,p)$$

It is somewhat unfortunate that COR and DOR are defined in terms of linear *independence*, while the EXIT functions and the probability of bit-error are defined in terms of linear *dependence*. This is primarily due to historical justifications: in order to agree with earlier definitions in the literature, they should be defined in this way.

It is important to remark that the definitions given here for the EXIT functions and for the probability of bit-error are, on the surface, very different than those given in the literature. However, these definitions turn out to be equivalent when dealing with the MEC; in the next section, we prove this equivalence for the probability of bit-error, and the equivalence for the EXIT functions is proven in Section 4.3.

2.2 Properties and Relationships

In this section we collect some basic facts about the measures defined above.

Proposition 2.1. *These measures can be equivalently defined in terms of expected differences in rank. Specifically,*

$$\begin{aligned}
\rho_i(A,p) &= \mathbb{E}(\mathrm{rank} A^{(i)}[p]) - \mathbb{E}(\mathrm{rank} A^{(i-1)}[p]) \\
\Psi_i(A,p) &= \mathbb{E}(\mathrm{rank} A[p]) - \mathbb{E}(\mathrm{rank} A_{\sim i}[p]) \\
p_j(1-\underline{h}_j(A,\underline{p})) &= \mathbb{E}(\mathrm{rank} A[\underline{p}]) - \mathbb{E}(\mathrm{rank} A[\underline{p}_{\underline{\sim} j}])
\end{aligned}$$

Proof. Beginning with the COR values, consider the quantity

$$\mathbb{E}(\mathrm{rank} A^{(i)}[p]) - \mathbb{E}(\mathrm{rank} A^{(i-1)}[p])$$

By adding the ith row, we will either keep the rank constant or increase it by 1, and the latter will happen if and only if the ith row is independent of the previous rows. Thus, the expected increase in rank is precisely the probability that row i is independent of the previous rows, namely $\rho_i(A,p)$. The proof for Ψ_i is nearly identical, the only difference being that we now consider all other rows rather than only the previous ones.

For $\underline{h}_j(A,\underline{p})$, the proof is very similar, but one has to be a bit more careful. By moving from $A[\underline{p}_{\underline{\sim} j}]$ to $A[\underline{p}]$, the rank will increase by 1 if and only if column j is chosen and is linearly independent of the other chosen columns, and the rank will stay the same otherwise. These two events—that column j is chosen and that it's independent of the other chosen columns—are independent, so the probability that both happen is simply the product of their probabilities, namely $p_j(1-\underline{h}_j(A,\underline{p}))$, where the 1 minus comes from the fact that $\underline{h}_j$ is defined in terms of linear dependence. □

Proposition 2.2. *There is a realtionship between DOR values and scalar probabilities of bit-error when we consider the removal of a column or the removal of a row from A, namely*

$$P_{e,j}(A_{\sim i}, p) - P_{e,j}(A, p) = \Psi_i(A, p) - \Psi_i(A[\sim j], p)$$

Proof. First, from the definition of the probability of bit-error, we know that

$$P_{e,j}(A_{\sim i}, p) - P_{e,j}(A, p) = p \cdot (h_j(A_{\sim i}, p) - h_j(A, p))$$

Now we proceed with a direct calculation from the previous proposition:

$$\begin{aligned} p(h_j(A_{\sim i}, p) - h_j(A, p)) &= p(1 - h_j(A, p)) - p(1 - h_j(A_{\sim i}, p)) \\ &= (\mathbb{E}(\operatorname{rank} A[p] - \operatorname{rank} A[\sim j][p])) - (\mathbb{E}(\operatorname{rank} A_{\sim i}[p] - \operatorname{rank} A_{\sim i}[\sim j][p])) \\ &= (\mathbb{E}(\operatorname{rank} A[p] - \operatorname{rank} A_{\sim i}[p])) - (\mathbb{E}(\operatorname{rank} A[\sim j][p] - \operatorname{rank} A_{\sim i}[\sim j][p])) \\ &= \Psi_i(A, p) - \Psi_i(A[\sim j], p) \end{aligned}$$

which is what we wanted to show. □

Proposition 2.3. *We can express the EXIT function as the derivative of the expected dimension of the nullspace of $A[\underline{p}]$:*

$$\underline{h}_j(\underline{p}) = \frac{\partial}{\partial p_j} \mathbb{E}(\dim \ker A[\underline{p}])$$

for all $j \in [n]$.

Proof. By the rank-nullity theorem, we know that the rank of a matrix plus the dimension of its nullspace is the number of columns. Applying expectations to this fact, we get that

$$\mathbb{E}(\dim \ker A[\underline{p}]) = \mathbb{E}(\#\text{columns of } A[\underline{p}] - \operatorname{rank} A[\underline{p}]) = \sum_{k=1}^{n} p_k - \mathbb{E}(\operatorname{rank} A[\underline{p}])$$

where we have used the fact that the expected number of columns is the expected number of erasures, which is just the sum over the probability that each bit is erased, namely $\sum p_k$. Differentiating this gives us

$$\begin{aligned} \frac{\partial}{\partial p_j} \mathbb{E}(\dim \ker A[\underline{p}]) &= \frac{\partial}{\partial p_j} \left(\sum_{k=1}^{n} p_k - \mathbb{E}(\operatorname{rank} A[\underline{p}]) \right) \\ &= 1 - \frac{\partial}{\partial p_j} \mathbb{E}(\operatorname{rank} A[\underline{p}]) \end{aligned}$$

To calculate this last derivative, suppose we increase the jth coordinate of $\underline{p}$ by some small value Δp_j. Then what is the difference $\mathbb{E}(\text{rank}A[\underline{p} + \Delta p_j]) - \mathbb{E}(\text{rank}A[\underline{p}])$? Note that since we are only changing the probability of selection of a single column, the rank can either stay the same or increase by 1. For it to increase by 1, all three of the following must happen:

- We must not have selected column j when picking with probabilities $\underline{p}$
- We must select column j when picking with probabilities $\underline{p} + \Delta p_j$
- Column j must be linearly independent of the other picked columns

What is the probability that all of these happen? The probability that the first two happen, i.e. that we didn't pick the column before but do now, is simply Δp_j. Thus,

$$\begin{aligned}\mathbb{E}(\text{rank}A[\underline{p} + \Delta p_j]) - \mathbb{E}(\text{rank}A[\underline{p}]) = \\ = \Delta p_j \cdot \mathbb{P}(\text{column } j \text{ is independent of the columns in } A[\underline{p}_{\sim j}])\end{aligned}$$

Dividing out by Δp_j and taking the limit as $\Delta p_j \to 0$ shows us that

$$\begin{aligned}\frac{\partial}{\partial p_j}\mathbb{E}(\text{rank}A[\underline{p}]) &= \mathbb{P}(\text{column } j \text{ is independent of the columns in } A[\underline{p}_{\sim j}]) \\ &= 1 - \underline{h}_j(\underline{p})\end{aligned}$$

Combining this with our earlier calculation gives exactly what we wanted. □

Recall that a linear code is one that is a k-dimensional vector subspace of the vector space $\mathbb{F}^n$, where $\mathbb{F}$ is some field. Any such subspace can be given as the nullspace of some $(n-k) \times n$ matrix, and such a matrix is called a parity-check matrix of the code. More formally,

Definition 2.5. Given an $[n, k]$ linear code $C \subseteq \mathbb{F}^n$, a *parity-check matrix* (PCM) for C is any $(n-k) \times k$ matrix A satisfying $Ax = 0$ for all $x \in C$. Note that the condition on the dimensions of A guarantees that A will have full row rank.

With this definition, we can state our final proposition.

Proposition 2.4. *Let A be as above, except that now we think of it as the PCM of a code C with blocklength n. We pick a codeword $x \in C$ uniformly at random, and let $y(p)$ denote the random vector that we get by erasing the jth bit of x with probability p. Then*

$$\begin{aligned} P_{e,j}(A,p) &= \mathop{\mathbb{P}}_{x,y(p)}(x_j \text{ cannot be recovered from } y(p)) \\ &= \mathop{\mathbb{P}}_{x,y(p)}(\exists x' \in C, x'_j \neq x_j, x' \text{ can yield } y(p) \text{ via erasures}) \end{aligned}$$

This justifies the name "probability of bit-error," and is in fact the usual definition for this quantity.

Proof. In order for us to make an error in decoding coordinate j, two things must happen. First, we must erase coordinate j, for otherwise we will certainly be able to recover it. And second, we must have some $x' \in C$ with $x'_j \neq x_j$ such that when we erase according to the erasure pattern given by $y(\underline{p})$, x and x' become indistinguishable. This means that the support of $x - x'$ must be contained in the erasure pattern; however, by linearity, $x - x'$ is a codeword, so if its support is in the erasure pattern, that must mean that the columns indexed by the erasure pattern are linearly dependent. Moreover, since x and x' disagree at the jth coordinate, there must be some such linear dependence that contains the jth column. In other words, column j must be linearly dependent on the other columns picked in $A[\underline{p}_{\sim j}]$.

Finally, observe that these two events are independent, so the probability that both happen is just the product of their probabilities, which is simply $ph_j(A,p)$, which was our definition for $P_{e,j}(A,p)$. □

2.3 Algebraic Measures of Tensor Power Matrices

Let $\mathbb{F}$ be any field, let n be a power of 2, and let G_n be the matrix over $\mathbb{F}$ defined by

$$G_n = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes \log n}$$

Note that the entries of this matrix are only 0's and 1's, and therefore this can be viewed as a matrix over any field.

Many important codes can be derived from G_n, and they are all based on the following simple idea. First, we pick some measure of "goodness" on the

rows of G_n. Then, we take the submatrix of G_n obtained by keeping only those rows which are the "best" under this metric. Finally, we define a code whose PCM is this matrix. Two important examples are the Reed-Muller codes defined in [14, 13], where goodness is measured by the weight of the rows, and more recently Arıkan's polar codes, defined in [3, 2], where goodness is measured by the entropy (or mutual information); both of these will soon be discussed in greater detail.

One important property of the tensor product matrices G_n is that we can explicitly calculate the values of our four algebraic independence measures on them, as shown in the following lemmas.

Lemma 2.1. *For any $j \in [n]$,*

$$\underline{h}_j(G_n, \underline{p}) = h_j(G_n, p) = P_{e,j}(G_n, p) = 0$$

Proof. Since G_n is an invertible square matrix, none of its columns is linearly dependent on any collection of other columns, so $\underline{h}_j(G_n, \underline{p})$ must be identically zero, which implies the same for h_j and $P_{e,j}$. □

Lemma 2.2. *Define the functions*

$$\ell(x) = 2x - x^2$$
$$r(x) = x^2$$

and define a branching process of depth $\log n$ *and offspring* 2 *(i.e., each node has exactly two descendants) as follows: the base node has value p, and for a node with value x, its left-hand child has value $\ell(x)$ and its right-hand child has value $r(x)$. Then the n leaf-nodes of this branching process are, in order, the values $\rho_i(G_n, p)$ for $1 \leq i \leq n$.*

Proof. The proof of this lemma is rather technical and can be found in [1]. One important thing to note, however, is that the functions ℓ and r do not depend on $\mathbb{F}$, while the COR values $\rho_i(G_n, p)$ are defined in terms of linear independence, and therefore do depend on $\mathbb{F}$, *a priori.* Thus, one consequence of this lemma is that the COR values of G_n are field-independent, though their definition does depend on the base field. □

For the final lemma, we will need some notation. Given a binary vector v, let $w(v)$ denote the Hamming weight of v, namely the number of non-zero coordinates in v. Similarly, for a positive integer i, let $w(i)$ denote the Hamming weight of the binary representation of i, or equivalently the minimum number of powers of 2 whose sum is i.

Lemma 2.3. *Fix $\mathbb{F} = \mathbb{F}_2$, the binary field. Then for any n a power of 2 and for all $i \in [n]$,*

$$\Psi_i(G_n, p) = p^{2^{w(i-1)}} = p^{n/w((G_n)_i)}$$

Proof. First, we will show that the two terms on the right-hand side are equal for all n and $1 \leq i \leq n$, i.e. that

$$2^{w(i-1)} = \frac{n}{w((G_n)_i)}$$

This is proved by induction on n. The base case is $n = 2$, where

$$G_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

and so

$$w((G_2)_1) = 2 \qquad\qquad w((G_2)_2) = 1$$

In addition,

$$2^{w(0)} = 1 \qquad\qquad 2^{w(1)} = 2$$

Thus, $2^{w(i-1)} = n/w((G_n)_i)$ is indeed true for $n = 2$ and $1 \leq i \leq 2$. For the inductive step, observe that we may write G_n in block form as

$$G_n = \begin{pmatrix} G_{n/2} & G_{n/2} \\ 0 & G_{n/2} \end{pmatrix}$$

Therefore, if $1 \leq i \leq n/2$, then $(G_n)_i$ is simply a concatenation of two copies of $(G_{n/2})_i$, and thus

$$w((G_n)_i) = 2w((G_{n/2})_i) = 2\frac{n/2}{2^{w(i-1)}} = \frac{n}{2^{w(i-1)}}$$

where the middle equality is the inductive hypothesis. Similarly, we see that for $n/2 + 1 \leq i \leq n$, we have that $(G_n)_i$ is a row of zeros concatenated to $(G_{n/2})_{i-n/2}$, and therefore

$$w((G_n)_i) = w((G_{n/2})_{i-n/2}) = \frac{n/2}{2^{w(i-n/2-1)}} = \frac{n}{2^{w(i-1)}}$$

where we have used the fact that since $i \in [n/2 + 1, n]$, the binary expansion of $i - 1$ has a 1 in the location corresponding to $n/2$, which means that $w(i -$

$n/2-1) = w(i-1)-1$. From this, we conclude that the two quantities on the right-hand side of the statement of the lemma are indeed equal.

So it only remains to prove that for all n and i, $\Psi_i(G_n, p)$ is equal to either $p^{2^{w(i-1)}}$ or to $p^{n/w((G_n)_i)}$. For this, we again proceed by induction. The base case is again G_2; we can see that in order for the first row of $G_2[p]$ to be independent of the second row, we must select either only the first column or both columns, meaning that

$$\Psi_1(G_2, p) = p(1-p) + p^2 = p$$

On the other hand, for the second row to be independent of the first, we must select both columns, so

$$\Psi_2(G_2, p) = p^2$$

as desired. For the inductive step, suppose that we have proven that for all $i \in [n/2]$,

$$\Psi_i(G_{n/2}, p) = p^{2^{w(i-1)}}$$

To prove the same for G_n, we begin with the case where $i \leq n/2$. Rather than computing $\Psi_i(G_n, p)$, we will actually compute $1 - \Psi_i(G_n, p)$, namely the probability that row i of $G_n[p]$ is linearly dependent on the other rows. Since we are computing linear dependence over $\mathbb{F}_2$, this is simply

$$1 - \Psi_i(G_n, p) = \mathbb{P}\left(\exists S \subseteq [n] \setminus \{i\} : \sum_{k \in S} (G_n)_k[p] = (G_n)_i[p] \right)$$

We will treat the left-hand side of the matrix and the right-hand side of the matrix differently. For this purpose, let $L = \{1, \ldots, n/2\}$ and $R = \{n/2+1, \ldots, n\}$; then the probability above can be rewritten as

$$\mathbb{P}\left(\exists S \subseteq [n] \setminus \{i\} : \sum_{k \in S} (G_n)_k[p][L] = (G_n)_i[p][L] \wedge \sum_{k \in S} (G_n)_k[p][R] = (G_n)_i[p][R] \right)$$

We have done nothing so far; all this says is that if some set of vectors sum to another vector, then in particular, the first few coordinates of the summands sum to the first few coordinates of the sum, and similarly for the last coordinates. However, since the bottom-left corner of G_n is all zeros, the above is the same as

$$\mathbb{P}\left(\exists S : \sum_{k \in S \cap [n/2]} (G_n)_k[p][L] = (G_n)_i[p][L] \ \wedge \ \sum_{k \in S} (G_n)_k[p][R] = (G_n)_i[p][R] \right)$$

All we have done is change the index set on the first sum; it suffices to consider only those summands in the top half of the matrix. However, now we can use a "cloning" trick, as follows. We claim that for any set $S' \subseteq [n/2] \setminus \{i\}$, there exists a set S with $S' \subseteq S \subseteq [n] \setminus \{i\}$ such that

$$\sum_{k \in S} (G_n)_k[p][R] = (G_n)_i[p][R]$$

To see this, simply set $S = S' \cup \{i + n/2\} \cup (S' + n/2)$, where

$$S' + n/2 = \{k + n/2 : k \in S'\}$$

In other words, S consists of S', along with a clone of every element of S' in the bottom half of the matrix, along with a clone of row i in the bottom half. Then S clearly satisfies the properties we want, since

$$\begin{aligned}\sum_{k \in S} (G_n)_k[p][R] &= \sum_{k \in S'} (G_n)_k[p][R] + \sum_{k \in S'} (G_n)_{k+n/2}[p][R] + (G_n)_{i+n/2}[p][R] \\ &= (G_n)_i[p][R]\end{aligned}$$

where we use the fact that on the right-hand side of the matrix, row k and row $k + n/2$ are identical, so all the terms but the last one cancel.

In other words, we have just shown that given any set of rows that yields a linear dependence on the left-hand side, we can extend it to a set of rows that yields a linear dependence on both sides of the matrix. This implies that above, when we considered the probability that two events happen, the second event is a consequence of the first, so we are simply calculating the probability

$$\mathbb{P}\left(\exists S' \subseteq [n/2] \setminus \{i\} : \sum_{k \in S'} (G_n)_k[p][L] = (G_n)_i[p][L]\right)$$

However, now that we are working wholly on the left-hand side, we see that this is the same as

$$\mathbb{P}\left(\exists S' \subseteq [n/2] \setminus \{i\} : \sum_{k \in S'} (G_{n/2})_k[p] = (G_{n/2})_i[p]\right)$$

which is simply $1 - \Psi_i(G_{n/2}, p)$. Thus, we have shown that for $1 \leq i \leq n/2$,

$$\Psi_i(G_n, p) = \Psi_i(G_{n/2}, p) = p^{2^{w(i-1)}}$$

as desired.

For $n/2+1 \leq i \leq n$, a similar inductive proof must exist; however, we have been unable to find it. Nonetheless, it turns out to be unnecessary. We again split into two cases; we will first deal with the case where $n/2+1 \leq i \leq n-1$. We wish to prove that

$$\Psi_i(G_n, p) = p^{n/w((G_n)_i)}$$

Note that the right-hand side does not actually depend on the row we have chosen, only on the weight of that row in the matrix G_n. In addition, it turns out that the left-hand side also doesn't depend on the specific row chosen. The reason is that the definition of Ψ_i is clearly invariant under permutations of columns, since a permutation of the coordinates of a collection of vectors does not change the linear dependence properties of that collection (since linear dependence is a coordinate-wise relation). However, any row of a given weight can be turned into another row of the same weight by a simple permutation of columns. So the statement that we wish to prove only depends on the weight of row i, and on nothing else. Moreover, for any $n/2+1 \leq i \leq n-1$, the weight of $(G_n)_i$ is equal to the weight of some row in the top half of the matrix, and for those rows we have already shown that this equality holds. Therefore, the theorem is true for $n/2+1 \leq i \leq n-1$.

Therefore, the only case remaining to be checked is the case $i = n$. In this case, we have to show that $\Psi_n(G_n, p) = p^n$. To do so, we claim two things: first, that $(G_n)_n$ is independent of the other $n-1$ rows, and second, that for any $P \subsetneq [n]$, $(G_n)_n[P]$ is linearly dependent on the other rows of $G_n[P]$. This suffices because the probability that we pick all the columns is precisely p^n, and this is the only way to get linear independence. The first claim is true because G_n has full rank, so there can be no linear dependence among its rows. For the second claim, we again proceed by induction on n; we already checked the base case above. For the inductive step, let $P \subsetneq [n]$. If $n \notin P$, then $(G_n)_n[P]$ is the all-zeros vector, which is trivially dependent on the other rows. Otherwise, there is some $n \neq k \notin P$. If $k \in \{1, \ldots, n/2\}$, the top-left $G_{n/2}$ has more rows than columns, meaning that there is some subset of its rows that sum to zero. We can now do another "cloning" trick like the one used above: we take these rows, together with their clones in the bottom half, together with the $(n/2)$th row, and these will sum to $(G_n)_n[P]$. On the other hand, if $k \in \{n/2+1, \ldots, n-1\}$, then we work only within the bottom-right $G_{n/2}$; by the inductive hypothesis, there is some linear combination of these rows that sums to $(G_{n/2})_{n/2}[P \cap R]$, and this linear combination works for G_n as well.

Therefore, $(G_n)_n[P]$ is dependent on the other rows of $G_n[P]$, as desired. □

Having proven all the necessary basic facts about these algebraic measures, we now turn their applications in the world of codes.

3 Polar Codes

In 2009, Arıkan published a groundbreaking paper, [3], in which he defined polar codes. These codes were the first explicit examples of codes that achieve the Shannon capacity on any channel, and thus answered the main question left open from Shannon's original paper [15]. These codes are defined by having a PCM which is obtained by selecting some rows of G_n, where the selection is based on an information-theoretic notion of independence. Arıkan's construction inspired the construction defined below (originally published in [1]), in which we sought to replace this information-theoretic measure by a linear-algebraic measure of independence, namely the COR values.

3.1 High-Girth Matrices

Before explaining the construction, we will need to understand the relationship between some coding-theoretic and linear-algebraic properties. First of all, we can express the coding-theoretic property of achieving capacity on the $MEC(p)$ as a linear-algebraic property of a matrix, as shown in the following lemma.

Lemma 3.1. *Fix a sequence of codes $\mathscr{C} = \{C_n\}_{n\geq 1}$, and let H_n be a PCM of C_n. Then $\mathscr{C}$ can successfully decode on the $MEC(p)$ if and only if $H_n[p]$ has linearly independent columns with high probability.*

Therefore, $\mathscr{C}$ is capacity-achieving on the $MEC(p)$ if and only if $H_n[p]$ has linearly independent columns with high probability and

$$\lim_{n\to\infty} \frac{\textit{number of rows of } H_n}{\textit{number of columns of } H_n} = p$$

Proof. The second statement follows from the first and the fact that the rate of C_n is precisely 1 minus the fraction of the number of rows to the number of columns of H_n. So it suffices to show the first statement, which is proved analogously to Proposition 2.4.

For that, observe that the probability of error of decoding C_n is precisely

$$\begin{aligned} P_e(C_n) &= \mathbb{P}_E(\exists x, x' \in C_n, x \neq x', x[E^c] = x'[E^c]) \\ &= \mathbb{P}_E(\exists z \in C_n, z \neq 0, z[E^c] = 0) \end{aligned}$$

where E is the erasure pattern of the $MEC(p)$, i.e., a random subset of $[n]$ obtained by picking each element with probability p. In the second equality, we have used linearity to deduce that $z = x - x'$ is also a codeword.

Note that E has the property that there exists a codeword $z \in C_n$ such that $z[E^c] = 0$ if and only if the columns indexed by E in H_n are linearly dependent. Indeed, assume first that there exists such a codeword z, where the support of z is contained in E. Since z is in the kernel of H_n, the columns of H_n indexed by the support of z must add up to 0, hence any set of columns that contains the support of z must be linearly dependent. Conversely, if the columns of H_n indexed by E are linearly dependent, then there exists a subset of these columns and a collection of coefficients in $\mathbb{F}$ such that this linear combination is 0, which defines the support of a codeword z. Hence,

$$P_e(C_n) = \mathbb{P}_E(H_n[E] \text{ has linearly dependent columns})$$

Therefore, P_e will tend to 0 if and only if the right-hand side tends to 0, which is precisely what we wanted to show. □

This lemma allows us to precisely characterize capacity-achieving codes in terms of a linear-algebraic property of their PCMs. Because of this, it is useful to give this property a name:

Definition 3.1. Fix a parameter $p \in [0,1]$. A sequence of matrices $\{H_n\}_{n \geq 1}$ where H_n has n columns and full row rank is called a *p-high-girth family* if $H_n[p]$ has linearly independent columns with high probability, and

$$\lim_{n \to \infty} \frac{\text{rank}(H_n)}{n} = p$$

When the p is understood, we will simply call such families *high-girth*.

Note that by concentration of measure, $H_n[p]$ will have very close to pn columns. Moreover, it will have roughly pn rows, since the definition guarantees that the fraction of rows to columns in H_n will tend to p. Therefore, the high-girth property can be thought of as roughly saying that when we randomly pick a square submatrix of H_n, we will get full column rank with high probability.

3.2 Examples of High-Girth Matrices

There are many explicit examples of high-girth families.

1. Most obviously, any time we find a sequence of capacity-achieving linear codes, their PCMs will form a high-girth family. However, from a coding-theoretic point of view, this observation is uninteresting—we hope to use our understanding of high-girth matrices to construct new codes, not vice versa.

2. Fix some number n and suppose that our alphabet field $\mathbb{F}$ has at least $n+1$ elements in it. Fix nonzero elements $x_1, \ldots, x_n \in \mathbb{F}$ and consider the matrix

$$V = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{k-1} & x_2^{k-1} & x_3^{k-1} & \cdots & x_n^{k-1} \end{pmatrix}$$

   Then this matrix has the property that *any* square submatrix has full rank. That is because any square submatrix is a $k \times k$ Vandermonde matrix generated by distinct elements, so it must have nonzero determinant, and therefore have full rank. Therefore, if $|\mathbb{F}| = \infty$, then this construction will give us a high-girth family. However, this example cannot be used to construct high-girth families over any finite field $\mathbb{F}$, since once $n+1 > |\mathbb{F}|$, this construction will fail, as we will then get repeated columns.

3. Working over the binary field $\mathbb{F}_2$, a family of matrices whose nth member is a $(pn + \varepsilon_n) \times n$ matrix whose entries are iid $Ber(\frac{1}{2})$ random variables will be p-high-girth for any $p \in (0,1)$, where $\varepsilon_n = \omega(1)$ and $\varepsilon_n = o(n)$. As this fact is well-known (see, e.g., [9, Section 3.2]), we will only sketch a proof. By concentration of measure and the iid-ness, we see that picking random columns from this random matrix is essentially the same as simply picking a random $(pn+\varepsilon_n) \times pn$ matrix. Then the probability that the first column is the zero vector is $2^{-pn-\varepsilon_n}$. The probability that the second column is in the span of the first is the probability that a random vector lies in a 1-dimensional subspace, namely $2^{-pn-\varepsilon_n+1}$. More generally, the probability that the kth column is in the span of the previous columns is

$2^{-pn-\varepsilon_n+k}$. Thus, by the union bound, the probability of a linear dependence among the columns is at most

$$\sum_{k=0}^{pn-1} 2^{-pn-\varepsilon_n+k} = 2^{-pn-\varepsilon_n} \sum_{k=0}^{pn-1} 2^k \leq 2^{-pn-\varepsilon_n} \cdot 2^{pn} \to 0$$

where we have used $\varepsilon_n = \omega(1)$. Moreover, since we also know that $\varepsilon_n = o(n)$, we see that the fraction of rows to columns in these matrices tends to pn, so they indeed form a high-girth family.

4. A new method for constructing high-girth matrices was developed in [1], and is described below.

3.3 Conditional-Rank Matrices

Recall Lemma 2.2, which said that the COR values $\rho_i(G_n, p)$ can be found as the leaves of a branching process initialized at p. A key property of the branching process in Lemma 2.2 is that it is a balanced process, meaning that the average value of the two children of a node with value x is x again:

$$\frac{\ell(x) + r(x)}{2} = \frac{(2x - x^2) + x^2}{2} = x$$

This means that a random walk on this tree that goes left or right with probability $\frac{1}{2}$ defines a martingale. Moreover, since $\rho_i(G_n, p)$ is a probability, we have that this martingale stays in $[0, 1]$. So by Doob's martingale convergence theorem [6], we must have that this martingale converges almost surely to its fixed points. Its fixed points are those x's for which $\ell(x) = r(x) = x$. The only points satisfying this are 0 and 1, so almost all values attained by this branching process approach either 0 or 1; this is a property that Arıkan called *polarization* in [3]. Moreover, since the process is balanced, we must have that

$$\sum_{i=1}^{n} \rho_i(G_n, p) = np$$

This, together with the polarization, implies that for any $\delta \in (0, \frac{1}{2})$,

$$\lim_{n\to\infty} \frac{|\{i \in [n] : \rho_i(G_n, p) > 1 - \delta\}|}{n} = p$$
$$\lim_{n\to\infty} \frac{|\{i \in [n] : \rho_i(G_n, p) < \delta\}|}{n} = 1 - p$$

The reason is simple: if almost all of the values are very close to either 0 or 1, but they must sum up to np, then roughly a p fraction of them should be close to 1 and roughly a $1-p$ fraction should be close to 0. In fact, the branching process defined by the functions $\ell(x)$ and $r(x)$ was already studied in [4], and this allows us to say much more about the speed at which this branching process polarizes:

Theorem 3.1 (Application of [4])**.** *For any n,*

$$\frac{|\{i \in [n] : \rho_i(G_n, p) > 1 - 2^{-n^{0.49}}\}|}{n} = p + o(1)$$
$$\frac{|\{i \in [n] : \rho_i(G_n, p) < 2^{-n^{0.49}}\}|}{n} = (1-p) + o(1)$$

Hence the theorem tells us that the above martingale polarizes very quickly: apart from a vanishing fraction, all $\rho_i(G_n, p)$'s are exponentially close to 0 or 1 as $n \to \infty$. With this in mind, we define the following.

Definition 3.2. Let n be a fixed power of 2, and let $p \in [0,1]$ be fixed. Let $I \subset [n]$ be the set of indices i for which $\rho_i(G_n, p) > 1 - 2^{-n^{0.49}}$, and let $m = |I|$. By Theorem 3.1, we know that $m = pn + o(n)$. Let $R_{n,p}$ denote the $m \times n$ submatrix of G_n gotten by selecting all the columns of G_n, but only taking those rows indexed by I. We call $R_{n,p}$ the *COR matrix* of size n with parameter p.

We will index the rows of $R_{n,p}$ by $i \in I$, rather than $k \in [m]$. The most important property of $R_{n,p}$ is expressed in the following theorem.

Theorem 3.2. *For any $p \in [0,1]$, $R_{n,p}[p]$ has full row rank (i.e. rank m) with high probability, as $n \to \infty$. In fact, $R_{n,p}[p]$ has full row rank with probability $1 - o(2^{-n^{0.48}})$.*

Proof. For $i \in I$, let B_i be the event that the ith row of $R_{n,p}[p]$ is linearly dependent on the previous rows. Note that if $R_{n,p}[p]$ has full rank, then no B_i is satisfied, while if $R_{n,p}[p]$ has non-full rank, then there must be some linear dependence in the rows, so at least one B_i will be satisfied. In other words, the event whose probability we want to calculate is simply the event $\bigcap_{i \in I} B_i^c$.

Note that in our notation, the ith row of $R_{n,p}$ is also the ith row of G_n. Therefore, for any $P \subseteq [n]$, the ith row of $R_{n,p}[P]$ is the ith row of $G_n[P]$. This means that any linear dependence between the ith row of $R_{n,p}[P]$ and the previous rows automatically induces a linear dependence between the ith row

of $G_n[P]$ and the previous $i-1$ rows, since the previous rows in $G_n[P]$ are a superset of the previous rows in $R_{n,p}[P]$. Since this is true for any set $P \subseteq [n]$, we see that

$$\begin{aligned}
\mathbb{P}(B_i) &= \mathbb{P}(\text{the } i\text{th row of } R_{n,p}[p] \text{ is dependent on the previous rows of } R_{n,p}[p]) \\
&\leq \mathbb{P}(\text{the } i\text{th row of } G_n[p] \text{ is dependent on the previous rows of } G_n[p]) \\
&= 1 - \rho_i(G_n, p) \\
&< 2^{-n^{0.49}}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{P}\left(\bigcap_{i\in I} B_i^c\right) &= 1 - \mathbb{P}\left(\bigcup_{i\in I} B_i\right) \\
&\geq 1 - \sum_{i\in I} \mathbb{P}(B_i) \\
&> 1 - \sum_{i=1}^{m} 2^{-n^{0.49}} \\
&= 1 - [pn + o(n)]2^{-n^{0.49}} \\
&= 1 - o\left(2^{-n^{0.48}}\right) \\
&\to 1 \text{ as } n \to \infty
\end{aligned}$$

□

Intuitively, this should imply that $R_{n,p}$ is a p-high-girth family: we have shown that each $R_{n,p}[p]$ has full row rank, so it should also have full column rank, and thus be high-girth. However, there is a slight subtlety stemming from the fact that $R_{n,p}[p]$ will generally have more columns than rows, so full row rank will not imply full column rank. However, this is an easily fixable problem: we define $R'_{n,p}$ to be the same as $R_{n,p}$, except that we also select $o(n)$ additional rows from G_n. Then $R'_n[p]$ will have rank strictly more than m whp, and we can therefore conclude, via another concentration of measure argument:

Theorem 3.3. *For any $p \in [0,1]$, $R'_{n,p}$ is p-high girth.*

Since the proof of Theorem 3.2 works independently of the base field $\mathbb{F}$, the same is true of Theorem 3.3. Thus, the COR matrix construction is fully deterministic and works over any field. From all of this discussion, we can finally get a coding-theoretic application:

Corollary 3.1. *The code family* $\{C_n\}$ *where* $C_n = \ker(R'_n)$ *is capacity-achieving on the* $MEC(p)$ *for any* $p \in [0,1]$ *and any alphabet field* $\mathbb{F}$.

Proof. By Theorem 3.3, R'_n is high-girth, which by Lemma 3.1 implies that $\{C_n\}$ achieves capacity on the $MEC(p)$. We call this family the COR codes. □

3.4 Relation to Polar Codes

It is perhaps surprising that in a section named "Polar Codes," these codes have not yet been discussed in any detail. The reason is primarily pragmatic—it is not necessary to know anything about polar codes in order to understand the above material. Nevertheless, they certainly should be mentioned, since there are at least two close relationships between COR codes and polar codes.

First and foremost is one of inspiration. As anyone familiar with polar codes will realize, the construction of COR codes directly mimics that of polar codes: in both cases, one defines a measure of some sort of statistical independence on the rows of G_n, proves that this measure polarizes, and concludes that selecting only the rows that fare best under this measure yields the PCM of a capacity-achieving code. Thus, COR codes can be seen as a sort of conceptual relative of polar codes, with all of our definition and proof ideas inspired by what Arıkan had done in the realm of polar codes.

Moreover, it turns out that COR and polar codes are even more closely related; specifically, in the special case of the binary erasure channel (i.e. working over $\mathbb{F}_2$), it turns out that COR codes and polar codes are identical. The reason is that in this special case, Arıkan's measure of goodness, namely the conditional entropy, turns out to follow the exact same branching process as the COR values, namely that given in Lemma 2.2; this is proven in [3, Proposition 5]. There are many ways to think about this equivalence, but one of them is to consider the COR construction as a linear-algebraic proof that polar codes achieve capacity on the BEC. For Lemma 3.1 implies that achieving capacity on the BEC is, fundamentally, a linear-algebraic property, so in particular, the fact that polar codes achieve capacity on the BEC is a linear-algebraic fact. Previously, however, the only proof that was known for this was Arıkan's far more general and non-algebraic proof that polar codes achieve capacity on any channel. So the COR construction can be thought of as a linear-algebraic proof of this important linear-algebraic fact.

4 Reed-Muller Codes

From now on, we will be working only over the binary field $\mathbb{F}_2$. Our main object of study will be Reed-Muller (RM) codes, defined by [14, 13]. There are several equivalent ways of defining these codes, but the one most useful for our purposes is the one that demonstrates their relation to the tensor power matrix G_n:

Definition 4.1. Let n be a power of 2 and let $0 \leq r \leq \log n$. The (n, r) Reed-Muller code is the code whose PCM is given by selecting only those rows of G_n whose weight is at least 2^r. This code has blocklength n and rate

$$\sum_{k=r}^{\log n} \binom{\log n}{k}$$

since G_n has precisely $\binom{\log n}{k}$ rows of weight exactly 2^k.

Reed-Muller codes are of enormous interest and have been extensively studied since they were defined in the 1950s. Therefore, it is extremely surprising that almost nothing was known about their effectiveness at transmitting on the BEC until 2015, when two teams, Kumar-Pfister [12] and Kudekar et. al [11], independently proved that RM codes achieve capacity on the $BEC(p)$ for all $p \in [0, 1]$. Both teams used essentially the same ingenious proof technique, one that relies on some fairly heavy mathematical machinery. However, by Lemma 3.1, we know that the statement that RM codes are capacity-achieving is, fundamentally, a linear-algebraic statement. So one might hope that their proof could be simplified so that it does not require any of the sophisticated tools they use.

In what follows, we present a purely linear-algebraic version of almost their entire proof. We have yet to succeed in reinterpreting one major step, though we have some partial progress leading to a linear-algebraic question. If this question can be answered, we will have a linear-algebraic proof that RM codes achieve capacity on the BEC.

4.1 The Linear-Algebraic Area Theorem

By definition, RM codes achieve capacity on the BEC if and only if the probability of error in decoding erasures tends to zero. For now, rather than trying

to prove that the probability of error P_e is small, we will only try to prove that the probability of bit-error $P_{e,j}$ is small for all j. Recall that by Proposition 2.4, $P_{e,j}$ deserves its name—though we defined it as an algebraic measure, it really does capture the probability that we make an error in decoding. Of course, $P_{e,j}$ only considers errors in decoding the jth coordinate, and therefore proving that $P_{e,j}$ is small is, *a priori,* a weaker statement than proving that P_e is small. However, it was shown in [10] that for RM codes, proving that $P_{e,j}$ is small implies that P_e is small as well.

By our definition of $P_{e,j}$, we see that if we can prove that $\underline{h}_j$ is small for RM codes in certain regimes of $\underline{p}$, then we will be able to deduce that the bit-error probability is small as well, and therefore that they achieve capacity on the $BEC(p)$. The way that this is shown in [12, 11] is using the following important theorem:

Theorem 4.1 (The Area Theorem)**.** *Let A be an $m \times n$ matrix which is the PCM of a Reed-Muller code C. Then for any $j \in [n]$,*

$$\int_0^1 h_j(A,p)\,\mathrm{d}p = \frac{n-m}{n} = r(C)$$

where $r(C)$ is the rate of C.

Proof. The fact that $r(C) = (n-m)/n$ is simply the statement that A is an $m \times n$ matrix. So the actual strength of the theorem comes from the first equality, which will follow from Proposition 2.3. Recall that it stated that

$$\underline{h}_j(A,\underline{p}) = \frac{\partial}{\partial p_j}\mathbb{E}(\dim\ker A[\underline{p}])$$

If we define a function

$$\varphi(p) = \mathbb{E}(\dim\ker A[p])$$

then Proposition 2.3 will imply that

$$\varphi'(p) = \sum_{j=1}^{n} h_j(p)$$

by the multivariate chain rule. Then integrating both sides from 0 to 1 gives

$$\sum_{j=1}^{n}\int_0^1 h_j(p)\,\mathrm{d}p = \varphi(1) - \varphi(0) = \dim\ker A = n-m$$

since A has full rank. Finally, we claim that for RM codes, all the EXIT functions are equal. This follows from an important symmetry property of RM codes, namely that for any $j_1, j_2 \in [n]$ there exists a permutation $\pi \in S_n$ with the properties

- $\pi(j_1) = j_2$
- for any codeword of C, if we permute its coordinates according to π, we will get another codeword in C

For a proof of this fact, see e.g. [8, Corollary 4].

In our linear-algebraic language, this says that π permutes the columns of A without changing its nullspace; therefore, notions of linear dependence among columns are unchanged. Thus, this and our definition of the EXIT function tell us that $h_{j_2}(A, p) = h_{j_2}(A^\pi, p)$, where A^π is A with its columns permuted according to π. However, since $\pi(j_1) = j_2$, we also see that $h_{j_2}(A^\pi, p) = h_{j_1}(A, p)$. Therefore, the EXIT functions h_{j_1} and h_{j_2} are equal for all j_1, j_2, and thus all of the EXIT functions are equal. Therefore, our equality above can be rewritten as

$$n - m = \sum_{j=1}^{n} \int_0^1 h_j(p)\,\mathrm{d}p = n \int_0^1 h_j(p)\,\mathrm{d}p$$

for any $j \in [n]$. Dividing by n gives what we claimed. □

From the Area Theorem, [12, 11] proceed by invoking a very general theorem, the Friedgut-Kalai theorem from [7], which allows them to demonstrate that as n grows, the EXIT functions $h_j(p)$ start to look more and more like step functions: they are extremely close to 0 below some threshold value of p, and then spike up suddenly and stay extremely close to 1 after this threshold. The Friedgut-Kalai theorem does not enable them to determine where this threshold is, which would be a problem, except that the Area Theorem saves the day: the integral of such a step function is precisely 1 minus the location of the threshold, so from the Area Theorem, this threshold must take place at $1 - r(C)$. This, in turn, means that below the threshold, we have that

$$P_{e,j} = p h_j(p)$$

must be very small, since $h_j(p)$ is very small. Thus, they conclude that RM codes can successfully transmit over the $BEC(p)$ at all rates larger than $1 - p$, and thus that RM codes achieve capacity on the BEC.

4.2 Ideas Towards a New Proof

The Friedgut-Kalai theorem is very general and powerful, and we have not been able to find a linear-algebraic alternative for it, even in this one special application. However, we believe we have the beginnings of a replacement, using DOR values.

The reason to study DOR values comes from the following idea. We are working in a PCM obtained by selecting some subset of the rows of G_n. In G_n, we understand the bit-error probabilities: they are all zero, by Lemma 2.1. Now, suppose we discard rows from G_n one at a time; if we could understand how the bit-error probabilities change at each step, we could hopefully understand how they behave once we've discarded all the "bad" rows. Moreover, by Proposition 2.2, we can interpret a change in bit-error probabilities when we delete a row as a change in DOR values when we delete a column; recall that the proposition stated that

$$P_{e,j}(A_{\sim i}, p) - P_{e,j}(A, p) = \Psi_i(A, p) - \Psi_i(A[\sim j], p)$$

In other words, we can think of DOR values as being a sort of dual to bit-error probabilities, under the duality between rows and columns. Moreover, we know precisely what $\Psi_i(G_n, p)$ is, thanks to Lemma 2.3. So it is not unreasonable to hope that one can understand how the DOR values change each time we delete a row of small weight from G_n, and use this understanding, along with the duality given by Proposition 2.2, to successfully bound $P_{e,j}(A, p)$, and thus prove that RM codes achieve capacity on the BEC. However, we have been unable to do this, and therefore leave it as an open question:

Question: What happens to Ψ_i when we delete a low-weight row from G_n? What happens when we do this again and again, deleting one row at a time? Can we get sufficiently sharp bounds to bound $P_{e,j}$ and conclude that RM codes achieve capacity on the BEC?

4.3 Appendix: Equivalence to the earlier proof

In this section, we explain why the various concepts and results studied in Section 4.1 are indeed equivalent to those presented in [12, 11]. This section is only an appendix, and the above proofs are complete on their own.

First, recall the following basic definitions of Information Theory:

Definition 4.2. For a random variable X on a finite set $\mathscr{X}$, we define its entropy to be

$$H(X) = -\sum_{a \in \mathscr{X}} P_X(a) \log P_X(a)$$

Given another random variable Y on some finite set $\mathscr{Y}$, we define the conditional entropy to be

$$H(X \mid Y) = \sum_{b \in \mathscr{Y}} H(X \mid Y = b) P_Y(b)$$

where here $H(X \mid Y = b)$ means the entropy, in the sense above, of the random variable $X \mid Y = b$.

Proposition 4.1. *Fix a code C and a column index j. Pick a codeword $x \in C$ uniformly at random and let $y(\underline{p})$ denote its output under the $BEC(p)$. Then*

$$\underline{h}_j(\underline{p}) = H(x_j \mid y_{\sim j}(\underline{p}))$$

Proof. For any given erasure pattern E, we have one of two options: either column j is linearly dependent on the columns indexed by E, or it is not. If it is, then we get a contribution to $\underline{h}_j(\underline{p})$ equal to the probability that E will occur as an erasure pattern; moreover, in this situation, the entropy of guessing x_j from $y_{\sim j}(\underline{p})$ will be 1, since column j being dependent precisely means that we cannot guess it from the other coordinates, by the proof of Lemma 2.4. So we will also get a contribution to $H(x_j \mid y_{\sim j}(\underline{p}))$ equal to the probability of E occurring. Similarly, if column j is independent of the others, then we will get a contribution of 0 to both sides of the equation: the left-hand side by definition, and the right-hand side by the fact that independence means that we *can* guess x_j from the other coordinates, and will thus have zero entropy. Thus, summing over all erasure patterns E gives us the desired result. □

In [12, 11], they use $H(x_j \mid y_{\sim j}(\underline{p}))$ as the definition of the EXIT function, since it is a more broadly-applicable information-theoretic definition. However, as we can see, it can really be thought of as a linear-algebraic quantity when dealing with the BEC.

Proposition 4.2. *Under the same conditions as the last proposition,*

$$\mathbb{E}(\dim \ker A[\underline{p}]) = H(x \mid y(\underline{p}))$$

Proof. We can write

$$H(x \mid y(\underline{p})) = \sum_{b \in \{0,1,?\}^n} \mathbb{P}(y(\underline{p}) = b) H(x \mid y(\underline{p}) = b)$$

For any observed vector b, we know that

$$H(x \mid y(\underline{p}) = b) =$$
$$= \begin{cases} 0 & x \text{ can be recovered from } b \\ \log \#\{z \in C : z \text{ can yield } b \text{ when erased}\} & \text{otherwise} \end{cases}$$

The reason is straightforward: if x can be recovered from b, then there is no entropy (since that is what recoverability means), whereas if x cannot be recovered from b, then we have a uniform distribution on all the possible codewords that can yield b, and its entropy is precisely log of the size of the support. This number above is, by linearity, the number of codewords whose supports are covered by the erasure pattern of b. This is the number of codewords in the nullspace of $A[E]$, where E is the erasure pattern. Therefore, its log is just $\dim \ker A[E]$, the dimension of this nullspace. Note that since $0 = \log 1$, this also covers the case when x can be uniquely recovered from b. Thus, we see that

$$\begin{aligned} H(x \mid y(\underline{p})) &= \sum_{b \in \{0,1,?\}^n} \mathbb{P}(y(\underline{p}) = b) H(x \mid y(\underline{p}) = b) \\ &= \sum_{b \in \{0,1,?\}^n} \mathbb{P}(y(\underline{p}) = b) \dim \ker A[E(b)] \\ &= \mathbb{E}(\dim \ker A[\underline{p}]) \end{aligned}$$

□

Note that these two propositions give us equivalent ways of expressing the quantities in Proposition 2.3. Thus, the way Proposition 2.3 is stated in [12, 11] is by saying that

$$\frac{\partial}{\partial p_j} H(x \mid y(\underline{p})) = H(x_j \mid y_{\sim i}(\underline{p}))$$

and similarly the Area Theorem is given as

$$\int_0^1 H(x_j \mid y_{\sim j}(p, \ldots, p)) \, \mathrm{d}p = H(x \mid y(p, \ldots, p)) \Big|_0^1 = \frac{n-m}{n}$$

With all this in mind, we see that our EXIT functions are the same as theirs, our Area Theorem is the same as theirs, and the proof presented in Section 4.1 is fundamentally the same as theirs.

References

[1] Emmanuel Abbe and Yuval Wigderson. "High-Girth matrices and polarization". In: *Information Theory (ISIT), 2015 IEEE International Symposium on*. June 2015, pp. 2461–2465. DOI: 10.1109/ISIT.2015.7282898.

[2] E. Arikan. "Source polarization". In: *2010 IEEE International Symposium on Information Theory*. June 2010, pp. 899–903. DOI: 10.1109/ISIT.2010.5513567.

[3] Erdal Arıkan. "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels". In: *Information Theory, IEEE Transactions on* 55.7 (2009), pp. 3051–3073.

[4] Erdal Arıkan and Emre Telatar. "On the rate of channel polarization". In: *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE. 2009, pp. 1493–1495.

[5] Stephan ten Brink. "Convergence of iterative decoding". In: *Electronics Letters* 35.10 (1999), pp. 806–808.

[6] J.L. Doob. *Stochastic Processes*. Wiley Publications in Statistics. John Wiley & Sons, 1953. ISBN: 9780471218135.

[7] Ehud Friedgut and Gil Kalai. "Every monotone graph property has a sharp threshold". In: *Proceedings of the American Mathematical Society* 124.10 (1996), pp. 2993–3002.

[8] Tadao Kasami, Shu Lin, and Wesley W Peterson. "New generalizations of the Reed-Muller codes—I: Primitive codes". In: *Information Theory, IEEE Transactions on* 14.2 (1968), pp. 189–199.

[9] V. F. Kolchin. *Random Graphs*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1999. ISBN: 9780521440813.

[10] Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D. Pfister, and Rüdiger L. Urbanke. "Comparing the Bit-MAP and Block-MAP Decoding Thresholds of Reed-Muller Codes on BMS Channels". In: *CoRR* abs/1601.06048 (2016). URL: http://arxiv.org/abs/1601.06048.

[11] Shrinivas Kudekar, Marco Mondelli, Eren Şaşoğlu, and Rüdiger L. Urbanke. "Reed-Muller Codes Achieve Capacity on the Binary Erasure Channel under MAP Decoding". In: *CoRR* abs/1505.05831 (2015). URL: http://arxiv.org/abs/1505.05831.

[12] Santhosh Kumar and Henry D. Pfister. "Reed-Muller Codes Achieve Capacity on Erasure Channels". In: *CoRR* abs/1505.05123 (2015). URL: http://arxiv.org/abs/1505.05123.

[13] D.E. Muller. "Application of Boolean algebra to switching circuit design and to error detection". In: *Electronic Computers, Transactions of the I.R.E. Professional Group on* EC-3.3 (Sept. 1954), pp. 6–12. ISSN: 2168-1740. DOI: 10.1109/IREPGELC.1954.6499441.

[14] I.S. Reed. "A class of multiple-error-correcting codes and the decoding scheme". In: *Information Theory, Transactions of the IRE Professional Group on* 4.4 (1954), pp. 38–49.

[15] Claude E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.