Introduction to Coding Theory **Técnico–ULisboa, 2026/2027**

Part 8: More alphabet-reduction techniques; Reed–Muller codes

Lecturer: João Ribeiro

Recommended reading: Essential Coding Theory, Sections 9.1, 9.3, and 13.2.

1 Introduction

In previous lectures we saw Reed–Solomon codes, which have optimal rate-distance tradeoff but require a large alphabet, along with code concatenation, which allows us to construct good codes over small alphabets from great codes over large alphabets. In this part we will look into other ways of constructing binary linear codes. This includes other alphabet-reduction techniques beyond code concatenation, and an extension of Reed–Solomon codes, called *Reed–Muller codes*.

2 Some more alphabet-reduction techniques

We have previously seen how code concatenation allows us to take a code over a larger alphabet with great properties and get from it a *binary* code with desirable properties. In this section we discuss some other natural approaches for extracting binary codes from codes over larger alphabets.

2.1 Subfield-subcodes

Suppose that you are given a linear code $\mathcal{C} \subseteq \mathbb{F}_{q^r}^n$ with q a prime power and $r \geq 1$ an integer. We may naturally see $\mathbb{F}_q$ as a subset of $\mathbb{F}_{q^r}$. Then, we can get a new code by keeping only the codewords of $\mathcal{C}$ with entries in the base field $\mathbb{F}_q$. More precisely, we define the $\mathbb{F}_q$*-subfield-subcode* $\mathcal{C}|_{\mathbb{F}_q}$ as

$$\mathcal{C}|_{\mathbb{F}_q} = \mathcal{C} \cap \mathbb{F}_q^n.$$

For simplicity, we will focus on the binary case where $q = 2$.

Clearly, $\mathcal{C}|_{\mathbb{F}_q}$ is a linear code with the same block length n as $\mathcal{C}$. Furthermore, the minimum distance of $\mathcal{C}|_{\mathbb{F}_q}$ is at least that of $\mathcal{C}$, and, if $\mathcal{C}$ has an efficient decoding algorithm, then that same algorithm can be used to decode $\mathcal{C}|_{\mathbb{F}_q}$. However, it is less clear how we can control the dimension of $\mathcal{C}|_{\mathbb{F}_q}$.

Taking the subfield-subcode of a large-alphabet linear code $\mathcal{C}$ sometimes yields very interesting small-alphabet linear codes. We present one such concrete example. Suppose that we are trying to obtain a binary linear code with some target minimum distance d (here we think of d as much smaller than the block length n, say constant), high rate/low redundancy, and efficient encoding and decoding. Considering subfield-subcodes of Reed–Solomon codes gives all of this, with redundancy that is optimal up to low-order terms.

More precisely, for given integers $r, d \geq 1$, set $n = 2^r - 1$ and consider the Reed–Solomon code $\mathcal{C}$ over $\mathbb{F}_{2^r}$ with evaluation set $\mathbb{F}_{2^r}^* = \mathbb{F}_{2^r} \setminus \{0\}$, block length $n = 2^r - 1$, and distance d. Then, the corresponding $\mathbb{F}_2$-subfield-subcode $\mathcal{C}|_{\mathbb{F}_2}$ has great properties. This code is usually called a (primitive, narrow-sense) *BCH code*, where "BCH" stands for "Bose-Chaudhuri-Hocquenghem", the three people who first studied these codes [Hoc59, BR60], and we denote it by $\mathsf{BCH}_{n,d}$.

Theorem 1 *Fix integers $r, d \geq 1$. Set $n = 2^r - 1$. Then, $\mathsf{BCH}_{n,d}$ is a binary linear code of block length n, minimum distance at least d, and dimension at least*

$$n - \left\lceil \frac{d-1}{2} \right\rceil \log(n+1).$$

Proof: You will prove this in the problem set. ∎

BCH codes are one of the most important family of binary linear codes. They inherit efficient decoders from Reed–Solomon codes, and their redundancy $\left\lceil \frac{d-1}{2} \right\rceil \log(n+1)$ is essentially optimal when d is small compared to n (but not when d is, say, linear in n).

2.2 Trace codes

Yet another way of obtaining a linear code over $\mathbb{F}_q$ from a linear code over a larger field $\mathbb{F}_{q^r}$ is by taking *traces*. The *trace function* Tr from $\mathbb{F}_{q^r}$ to $\mathbb{F}_q$ is given by

$$\mathrm{Tr}(x) = x + x^q + \cdots + x^{q^{r-1}}.$$

Consider a linear code $\mathcal{C} \subseteq \mathbb{F}_{q^r}^n$. Then, its corresponding *trace code* is

$$\mathrm{Tr}(\mathcal{C}) = \{(\mathrm{Tr}(c_1), \mathrm{Tr}(c_2), \ldots, \mathrm{Tr}(c_n)) : c \in \mathcal{C}\}.$$

We begin by recording some basic properties of the trace function.

Lemma 1 *The trace function satisfies the following properties:*

- $\mathrm{Tr}(\alpha \cdot x + y) = \alpha \,\mathrm{Tr}(x) + \mathrm{Tr}(y)$ *for all* $\alpha \in \mathbb{F}_q$ *and* $x, y \in \mathbb{F}_{q^r}$.
- $\mathrm{Tr}(x) \in \mathbb{F}_q$ *for all* $x \in \mathbb{F}_{q^r}$.
- $|\{\beta \in \mathbb{F}_{q^r} : \mathrm{Tr}(\beta) = \alpha\}| = q^{r-1}$ *for all* $\alpha \in \mathbb{F}_q$.

Proof: You will prove this in the problem set. ∎

From Lemma 1 it is not hard to see that $\mathrm{Tr}(\mathcal{C}) \subseteq \mathbb{F}_q^n$ and that $\mathrm{Tr}(\mathcal{C})$ is linear. However, it is less clear whether we can generally control the dimension and distance of $\mathrm{Tr}(\mathcal{C})$ in terms of those of $\mathcal{C}$.

We will now see that we can get very interesting binary codes by taking the trace of an appropriate Reed–Solomon code. Take $\mathcal{C}$ to be a Reed–Solomon code over $\mathbb{F}_{2^r}$ with full evaluation set $\mathbb{F}_{2^r}$ and even dimension k. The trace code $\mathrm{Tr}(\mathcal{C}) \subseteq \mathbb{F}_2^n$ is called a *dual BCH code*. This name is justified by the fact that there is a connection between subfield-subcodes and trace codes, captured by the following result of Delsarte.

Theorem 2 (Delsarte) *Fix an arbitrary linear code $\mathcal{C} \subseteq \mathbb{F}_{q^r}^n$. Then,*

$$(\mathcal{C}|_{\mathbb{F}_q})^\perp = \mathrm{Tr}(\mathcal{C}^\perp).$$

Proof: You will prove this in the problem set. ∎

Since the dual of a Reed–Solomon code with full evaluation set is also a Reed–Solomon code (you will also prove this in the problem set), we get that $\mathrm{Tr}(\mathcal{C})$ is the dual of the $\mathbb{F}_2$-subfield-subcode of a Reed–Solomon code with full evaluation set (a type of extended BCH code).

We now discuss the dimension and distance of $\mathrm{Tr}(\mathcal{C})$ without getting into full details. If the starting Reed–Solomon code $\mathcal{C}$ has block length $n = 2^r$ and dimension k, then $\mathrm{Tr}(\mathcal{C})$ has dimension at most $k \log n$. We will see that the dimension of $\mathrm{Tr}(\mathcal{C})$ is not much smaller than this.

First, observe that

$$\mathrm{Tr}(x^2) = x^2 + (x^2)^2 + \cdots + (x^2)^{2^{r-1}} = x^2 + x^4 + \cdots + x^{2^{r-1}} + x = \mathrm{Tr}(x),$$

since $\alpha^{2^r} = \alpha$ for all $\alpha \in \mathbb{F}_{2^r}$. More generally, we have

$$\mathrm{Tr}(\alpha x^{2j}) = \mathrm{Tr}(\alpha^{2^{r-1}} x^j)$$

for all $\alpha, x \in \mathbb{F}_{2^r}$. We can use this to conclude that every codeword $c \in \mathrm{Tr}(\mathcal{C})$ can be written as

$$c = (\mathrm{Tr}(f(x)))_{x \in \mathbb{F}_{2^r}}$$

for some polynomial $f \in \mathbb{F}_{2^r}[x]$ of degree at most $k-1$ *and which only contains monomials of odd degree, plus a constant term.*

As we will see, when $k \leq 2^{r/2} = \sqrt{n}$, any two such polynomials f and g with $f - g$ non-constant satisfy $(\mathrm{Tr}(f(x)))_{x \in \mathbb{F}_{2^r}} \neq (\mathrm{Tr}(g(x)))_{x \in \mathbb{F}_{2^r}}$. If $f = \alpha + g$ for some $\alpha \in \mathbb{F}_{2^r}$, then, by Lemma 1, the traces coincide for exactly half of the choices of α. Since there are $2^{r \cdot \lceil \frac{k-1}{2} \rceil}$ choices for the odd degree monomials, we get that $\mathrm{Tr}(\mathcal{C})$ has dimension

$$1 + \left\lceil \frac{k-1}{2} \right\rceil \log n.$$

We now turn to the distance of $\mathrm{Tr}(\mathcal{C})$. Since this is a linear code, it suffices to bound its minimum weight. This entails understanding the number of zeros of the maps $x \mapsto \mathrm{Tr}(f(x))$ for polynomials $f \in \mathbb{F}_{2^r}[x]$ of degree at most $k-1$ with only odd-degree monomials. We can write the weight of $c = (\mathrm{Tr}(f(x)))_{x \in \mathbb{F}_{2^r}}$ in terms of the exponential sum

$$S(f) = \sum_{x \in \mathbb{F}_{2^r}} (-1)^{\mathrm{Tr}(f(x))}$$

since

$$S(f) = |\{x : \mathrm{Tr}(f(x)) = 0\}| - |\{x : \mathrm{Tr}(f(x)) = 1\}| = 2^r - 2 \cdot w_H(c) = 2\left(\frac{n}{2} - w_H(c)\right)$$

Since $x \mapsto (-1)^{\mathrm{Tr}(x)}$ is a character[1] of $\mathbb{F}_{2^r}$, the *Weil–Carlitz–Uchiyama* bound [CU57] allows us to control $|S(f)|$.

[1] A function $\psi : \mathbb{F}_q \to \mathbb{C}$ is a *character* of $\mathbb{F}_q$ if $\psi(x+y) = \psi(x) \cdot \psi(y)$ for all $x, y \in \mathbb{F}_q$.

Lemma 2 (Weil–Carlitz–Uchiyama) *Let $f \in \mathbb{F}_{2^r}[x]$ be a polynomial of degree at most $k-1$ that is not of the form $g(x)^2 + g(x) + b$ for any $g \in \mathbb{F}_{2^r}[x]$ and $b \in \mathbb{F}_{2^r}$. Then,*

$$|S(f)| \leq (k-2)2^{r/2}.$$

Note that our polynomials f satisfy the hypothesis of Lemma 2, since they have odd degree, and that the bound is much better than the naive one, which is $2^r = n$ (so a lot of cancellations occur in the sum $S(f)$). The discussion above yields the following parameters of dual BCH codes. In sum, dual BCH codes have dimension $\Theta(k \log n)$ and large relative distance $\frac{1-\varepsilon}{2}$ with $\varepsilon \approx k/\sqrt{n}$.

Theorem 3 *When $k \leq 2^{r/2} = \sqrt{n}$, the dual BCH code $\mathrm{Tr}(\mathcal{C})$ above has block length $n = 2^r$, dimension $1 + \left\lceil \frac{k-1}{2} \right\rceil \log n$, and minimum distance at least*

$$\frac{n}{2}\left(1 - \frac{k-2}{\sqrt{n}}\right).$$

3 Reed–Muller codes

We now present a flexible family of polynomial-based linear codes which generalizes some of the codes we have seen before, both over $\mathbb{F}_2$ (the Hadamard code) and over large alphabets (Reed–Solomon codes).

We denote by $\mathbb{F}_q[x_1, \ldots, x_m]$ the set of m-variate polynomials over $\mathbb{F}_q$. These are functions of the form

$$f(x_1, \ldots, x_m) = \sum_{i_1, \ldots, i_m \geq 0} \alpha_{i_1, \ldots, i_m} x_1^{i_1} \cdots x_m^{i_m},$$

where $\alpha_{i_1,\ldots,i_m} \in \mathbb{F}_q$. We call $x_1^{i_1} \cdots x_m^{i_m}$ a *monomial*, and define its total degree to be $i_1 + \cdots + i_m$. The total degree of f, denoted by $\deg f$, is the largest total degree attained by a monomial $x_1^{i_1} \cdots x_m^{i_m}$ for which $\alpha_{i_1,\ldots,i_m} \neq 0$.

Reed–Muller codes correspond to evaluations of bounded-degree multivariate polynomials over $\mathbb{F}_q$. They were first introduced by Reed [Ree54], and then Muller [Mul54] gave an efficient decoding algorithm correcting the optimal number of errors. More precisely, given a field $\mathbb{F}_q$, a number of variables m, and degree bound r, we define the Reed–Muller code

$$\mathsf{RM}_{q,m,r} = \{(f(x_1, \ldots, x_m))_{(x_1,\ldots,x_m) \in \mathbb{F}_q^m} : f \in \mathbb{F}_q[x_1, \ldots, x_m], \deg f \leq r\}.$$

It is easy to see that $\mathsf{RM}_{q,m,r}$ is a q-ary linear code of block length $n = q^m$. Note that $\mathsf{RM}_{q,m=1,r}$ is a Reed–Solomon code with full evaluation set $\mathbb{F}_q$, while, at the other extreme, $\mathsf{RM}_{q=2,m,r=1}$ is the Hadamard code we saw before.

The dimension and minimum of distance of $\mathsf{RM}_{q,m,r}$ do not have general clean expressions. Instead, different arguments are required depending on how large r is compared to q. In our discussion we will focus on *binary* Reed–Muller codes, so we take $q = 2$. In the problem set you will work out the parameters for the complementing "low-degree" regime where $r < q$.

In the binary case, we may write an m-variate polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_m]$ in the simplified form

$$f(x) = \sum_{I \subseteq [m]} \alpha_I x^I,$$

where we shorthand $[m] = \{1, \ldots, m\}$ and $x^I = \prod_{i \in I} x_i$. The degree of f is then

$$\deg f = \max\{|I| : \alpha_I \neq 0\} \leq m.$$

We now proceed to compute the dimension and minimum distance of $\mathsf{RM}_{2,m,r}$. First, we claim that the dimension is $k = \sum_{i=0}^{r} \binom{m}{i} = \mathrm{Vol}_2(m, r)$. This is because the dimension equals the number of monomials of degree at most r. In turn, by the simplified form above, these correspond to sets $I \subseteq [m]$ of size at most r.

To control the minimum distance of $\mathsf{RM}_{2,m,r}$ we need to understand the number of zeros of multivariate polynomials with bounded degree.

Lemma 3 *Let $f \in \mathbb{F}_2[x_1, \ldots, x_m]$ be an arbitrary nonzero polynomial with $\deg f = r$. Then,*

$$|\{x \in \mathbb{F}_2^m : f(x) \neq 0\}| \geq 2^{m-r}.$$

Furthermore, there exists a degree-r polynomial f for which equality holds above.

Proof: We prove this lemma by induction on the number of variables m. Consider sampling $a = (a_1, \ldots, a_m)$ uniformly at random from $\mathbb{F}_2^m$. Then,

$$|\{x \in \mathbb{F}_2^m : f(x) \neq 0\}| = 2^m \cdot \Pr_a[f(a) \neq 0],$$

so we focus on showing that $\Pr_a[f(a) \neq 0] \geq 2^{-r}$.

For the base case $m = 1$ we may have either $r = 0$, in which case $f(x) = 1$ (and so $\Pr[f(a) \neq 0] = 1 = 2^{-0}$), or $r = 1$, in which case $f(x) = x$ or $f(x) = 1 + x$ (and so $\Pr[f(a) \neq 0] = 1/2 = 2^{-1}$).

For the induction step, we assume that the claim is true for all nonzero $(m-1)$-variate polynomials. Fix an arbitrary nonzero polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_m]$ of degree r. Then, define the $(m-1)$-variate polynomials

$$f_b(x_1, \ldots, x_{m-1}) = f(x_1, \ldots, x_{m-1}, b)$$

for $b \in \mathbb{F}_2$. Note that we may write

$$f(x) = f_0(x) + x_m \cdot (f_0(x) + f_1(x)).$$

Then,

$$\begin{aligned}
&\Pr_a[f(a) \neq 0] \\
&= \Pr[a_m = 0] \cdot \Pr[f_0(a_1, \ldots, a_{m-1}) \neq 0] + \Pr[a_m = 1] \cdot \Pr[f_1(a_1, \ldots, a_{m-1}) \neq 0] \\
&= \frac{1}{2} \Pr[f_0(a_1, \ldots, a_{m-1}) \neq 0] + \frac{1}{2} \Pr[f_1(a_1, \ldots, a_{m-1}) \neq 0].
\end{aligned}$$

First, assume that both f_0 and f_1 are nonzero polynomials. Then, the induction hypothesis gives that both probabilities in the sum above are at least 2^{-r}, and so $\Pr[f(a) \neq 0] \geq 2^{-r}$, as desired. But it may happen that one of f_0 or f_1 are the zero polynomial (but not both). If $f_0 \equiv 0$, then $f = x_m f_1$, and so $\deg f_1 \leq r-1$. Otherwise, if $f_1 \equiv 0$, then $f = (1+x_m)f_0$, and so $\deg f_0 \leq r-1$. So, by the induction hypothesis, if $f_{1-b} \equiv 0$, then f_b is nonzero and $\deg f_b \leq r-1$, so

$$\Pr[f(a) \neq 0] = \frac{1}{2}\Pr[f_b(a_1, \ldots, a_{m-1}) \neq 0] \geq \frac{1}{2} \cdot 2^{-(r-1)} = 2^{-r},$$

again as desired.

We leave the sharpness of this bound as an exercise. ■

Since $\mathsf{RM}_{2,m,r}$ is a binary linear code, its minimum distance equals its minimum weight. Every codeword is of the form $(f(x))_{x \in \mathbb{F}_2^m}$ for some $f \in \mathbb{F}_2[x_1, \ldots, x_m]$ of degree at most r. Therefore, Lemma 3 implies that the minimum distance of $\mathsf{RM}_{2,m,r}$ is at least 2^{m-r}.

The discussion above is summarized in the following theorem.

Theorem 4 *For $r \leq m$, $\mathsf{RM}_{2,m,r}$ is a binary linear code of block length $n = 2^m$, dimension $k = \sum_{i=0}^{r} \binom{m}{i}$, and minimum distance 2^{m-r}.*

In particular, we can set up the parameters of $\mathsf{RM}_{2,m,r}$ so that, for any $\tau \in (0, 1/2)$, it has rate approximately $n^{-(1-h(\tau))}$ and relative distance $n^{-\tau}$, but we can never get an asymptotically good code. Nevertheless, Reed–Muller codes are extremely important objects in coding theory:

- They have been used in deep-space and wireless communications.
- It turns out that they perform great against random errors. More precisely, Reed–Muller codes are known to achieve capacity over symmetric memoryless channels. This was conjectured for a long time, but it was only proved about a decade ago in a breakthrough result [KKM+16] first for erasure channels, and later generalized.
- Reed–Muller codes have also found important applications in other areas, such as complexity theory, cryptography, and sparse recovery.

If you would like to learn more than what is covered in these brief notes, check out the excellent survey of Abbe, Sberlo, Shpilka, and Ye [ASSY23].

4 Efficient decoding of binary Reed–Muller codes: Majority Logic decoding

By Theorem 4, we know that we can in principle correct any $t < \frac{1}{2} \cdot 2^{m-r}$ errors in an $\mathsf{RM}_{2,m,r}$ code. In this section we analyze the efficient *majority logic* decoding algorithm for this task, first proposed by Reed [Ree54].

The setup is as follows. The decoder is given as input a string $y \in \mathbb{F}_2^n$ with the promise that there is a codeword $c \in \mathsf{RM}_{2,m,r}$ such that $d(c,y) \le t < \frac{1}{2} \cdot 2^{m-r}$. Our goal is to find c. As we shall see, it is then possible to efficiently extract the encoded message from c.

Here, "efficient" means "polynomial in the block length n", and $n = 2^m$. This means that we are allowed to do things like iterate over subsets of $[m] = \{1, \ldots, m\}$.

The codeword $c \in \mathsf{RM}_{2,m,r}$ is of the form $c = (f(x))_{x \in \mathbb{F}_2^m}$ for some polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_m]$ of degree at most r. In turn, we may interpret y as the evaluations of some function $g : \mathbb{F}_2^m \to \mathbb{F}_2$, i.e., $y = (g(x))_{x \in \mathbb{F}_2^m}$. The guarantee that $d(y,c) \le t$ means that $f(x) \neq g(x)$ for at most t inputs x.

It will be useful to separate the monomials of f into "levels". We may write

$$f(x) = \sum_{I \subseteq [m]} \alpha_I x^I = f_0(x) + f_1(x) + \cdots + f_r(x),$$

where $f_i(x)$ collects the degree-i monomials of f, i.e.,

$$f_i(x) = \sum_{I \subseteq [m] : |I| = i} \alpha_I x^I.$$

We will also write $f_{\le i} = f_0 + f_1 + \cdots + f_i$.

At a very high level, the algorithm works by operating on restrictions of g to *subcubes*[2] of $\mathbb{F}_2^m$ in order to recover the coefficients of all maximum-degree monomials of f, i.e., we determine f_r. We then obtain a new function $g_{r-1} = g - f_r$. Since $f - f_r = f_{\le r-1}$, we get the guarantee that $g_{r-1}(x) \neq f_{\le r-1}(x)$ for at most t inputs x. We can then repeat the process with g_{r-1} in place of g to recover all maximum-degree monomials of $f_{\le r-1}$, i.e., we recover f_{r-1}. And so on until we recover all monomials of f.

We now dive deeper. The algorithm hinges on the following key lemma, which explains how we can recover coefficients of maximum-degree monomials of an m-variate polynomial f from evaluations of f on an appropriate subcube of $\mathbb{F}_2^m$.

Lemma 4 *Let $f \in \mathbb{F}_2[x_1, \ldots, x_m]$ be an arbitrary polynomial of degree at most r. For an arbitrary subset $T \subseteq [m]$ and a vector $v \in \mathbb{F}_2^{|T|}$, denote by $f_{T,v}$ the restriction of f to the subcube $\{x \in \mathbb{F}_2^m : x_T = v\}$. In other words, $f_{T,v}$ corresponds to fixing the coordinates indexed by T to v in f. Then, for every subset $S \subseteq [m]$ of size $|S| = r$ and $v \in \mathbb{F}_2^{m-r}$ it holds that*

$$\sum_{u \in \mathbb{F}_2^T} f_{\overline{S},v}(u) = \alpha_S,$$

where $\overline{S} = [m] \setminus S$ is the complement of S.

Proof: First, note that no degree-r monomial appears in $f_{\overline{S},v}$ except possibly for x^S, which has coefficient α_S in $f_{\overline{S},v}$ for all v.

[2] A *subcube* of $\mathbb{F}_2^m$ is a set obtained by fixing a subset of coordinates. More precisely, these are sets of the form $\{x \in \mathbb{F}_2^m : x_T = v\}$ for some subset $T \subseteq [m]$ and vector $v \in \mathbb{F}_2^{|T|}$, where $x_T = (x_i)_{i \in T}$.

Therefore, we may write

$$f_{\overline{S},v}(u) = \alpha_S u^S + (f_{\leq r-1})_{\overline{S},v}(u),$$

and so

$$\sum_{u \in \mathbb{F}_2^r} f_{\overline{S},v}(u) = \alpha_S + \sum_{u \in \mathbb{F}_2^r} (f_{\leq r-1})_{\overline{S},v}(u).$$

We claim that $\sum_{u \in \mathbb{F}_2^r} (f_{\leq r-1})_{\overline{S},v}(u) = 0$, in which case the desired result follows. To see this, note that for every monomial that appears in $(f_{\leq r-1})_{\overline{S},v}$ there are an even number of assignments $u \in \mathbb{F}_2^r$ that make it evaluate to 1, since the degree of such a monomial is at most $r-1$. Therefore, every such monomial leads to an even number of 1s in that sum. ■

We now show, using Lemma 4, that applying the same strategy to the noisy function g still allows us to extract the correct coefficient of f.

Lemma 5 *Let $f = \sum_{I \subseteq [m]} \alpha_I x^I$ be an arbitrary polynomial of degree at most r and suppose that $g : \mathbb{F}_2^m \to \mathbb{F}_2$ is such that $g(x) \neq f(x)$ for at most $t < \frac{1}{2} \cdot 2^{m-r}$ inputs x. Then, for any subset $S \subseteq [m]$ of size r it holds that*

$$\mathrm{Maj}\left(\left(\sum_{u \in \mathbb{F}_2^r} g_{\overline{S},v}(u) \right)_{v \in \mathbb{F}_2^{m-r}} \right) = \alpha_S.$$

Proof: By Lemma 4, if $g_{\overline{S},v}(u) = f_{\overline{S},v}(u)$ for all $u \in \mathbb{F}_2^r$, then

$$\sum_{u \in \mathbb{F}_2^r} g_{\overline{S},v}(u) = \sum_{u \in \mathbb{F}_2^r} f_{\overline{S},v}(u) = \alpha_S.$$

Since $g(x) \neq f(x)$ at fewer than $\frac{1}{2} \cdot 2^{m-r}$ inputs x, we get that this must hold for strictly more than $\frac{1}{2} \cdot 2^{m-r}$ of the v's. Since there are 2^{m-r} choices for v, we conclude that evaluating the sums $\sum_{u \in \mathbb{F}_2^r} g_{\overline{S},v}(u)$ for all $v \in \mathbb{F}_2^{m-r}$ and taking majority over all the outcomes yields the correct coefficient α_S. ■

We are now ready to describe the full algorithm on input $(g(x))_{x \in \mathbb{F}_2^m}$, with the promise that $g(x) \neq f(x)$ for fewer than $\frac{1}{2} \cdot 2^{m-r}$ inputs x, and $f(x) = \sum_{I \subseteq [m]} \alpha_I x^I$ is of degree at most r.

1. For all sets $S \subseteq [m]$ of size r, compute

$$\mathrm{Maj}\left(\left(\sum_{u \in \mathbb{F}_2^r} g_{\overline{S},v}(u) \right)_{v \in \mathbb{F}_2^{m-r}} \right) = \alpha_S.$$

   Correctness is guaranteed by Lemma 5.

2. Compute $f_r(x) = \sum_{S \subseteq [m] : |S| = r} \alpha_S x^S$.

3. If $r \geq 1$, go back to Step 1 with $(g(x) - f_r(x))_{x \in \mathbb{F}_2^m}$ in place of $(g(x))_{x \in \mathbb{F}_2^m}$ and $r-1$ in place of r.

4. We have now recovered $f = f_0 + f_1 + \cdots + f_r$.

5 Local error-correction

One of the reasons Reed–Muller codes are so important is that they support local error-correction procedures. Informally, suppose that we receive a corrupted $y \in \mathbb{F}_q^n$ obtained by introducing up to t errors in a Reed–Muller codeword $c \in \mathsf{RM}_{q,m,r}$, and that we wish to recover the i-th codeword symbol c_i. Then, it turns out we can do so with good probability by only checking a small number of symbols of y.

Codes with this property are said to be *locally-correctable*. We define it more formally now.

Definition 1 (Locally-correctable code) *We say that a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is $(\ell, \delta, \varepsilon)$-*locally-correctable *if there exists a* randomized[3] *decoder* Dec *that, given $y \in \mathbb{F}_q^n$ obtained by introducing any $t < \delta n$ errors in some codeword $c \in \mathcal{C}$ and a coordinate $i \in [n]$, queries y in at most ℓ coordinates and outputs c_i with probability at least $1 - \varepsilon$.*

Locally-correctable codes (LCCs) and their closely related counterparts, locally-decodable codes (LDCs)[4] have found many important applications in complexity theory, sublinear algorithms, and cryptography, among others. If you are interested in reading more about them, see Yekhanin's excellent monograph [Yek12].

There are many parameter tradeoffs in LCCs that are worth studying. Here we will focus on LCCs for constant fraction of errors δ, moderate (non-trivial) error probability $\varepsilon < 1/2$, and a very small number of queries ℓ. If needed, we can bring the error probability down quickly by repeatedly applying the local decoder a few times (each time with fresh randomness) and taking majority.

5.1 2-query LCCs

We begin by discussing the extreme case where we are only allowed $\ell = 2$ queries. In this setting, we will give a local decoder for the Hadamard code. Recall that the Hadamard code maps a message $x \in \mathbb{F}_2^k$ to the codeword

$$\mathsf{Enc}(x) = (\langle x, v \rangle)_{v \in \mathbb{F}_2^k} \in \mathbb{F}_2^{2^k}.$$

It is a linear code with block length $n = 2^k$ and minimum distance $n/2 = 2^{k-1}$.

Theorem 5 *The Hadamard code is an $(\ell = 2, \delta, \varepsilon = 2\delta)$-LCC for any error fraction $\delta < 1/4$.*

Proof: We describe the local decoder and leave its proof of correctness as an exercise for the problem set. We see the coordinates of vectors in $\mathbb{F}_2^{2^k}$ as being indexed by elements of $\mathbb{F}_2^k$. On input $y \in \mathbb{F}_2^{n=2^k}$ satisfying $d(y, c) < \delta n$ for some codeword c and a coordinate $v \in \mathbb{F}_2^k$, the decoder proceeds as follows:

[3] It is a good exercise for you to reflect on why randomness is needed.

[4] A code is said to be *locally-decodable* if we can recover the i-th bit of the *message* by querying a small number of coordinates of y. In contrast, in LCCs we wish to recover the i-th bit of the codeword. For codes that have the message embedded in the corresponding codeword, local correction is harder than local decoding.

1. Sample z uniformly at random from $\mathbb{F}_2^k$.
2. Query y_z and y_{v+z}.
3. Output the guess $\tilde{c}_v = y_z + y_{v+z} \pmod 2$.

It is not hard to show that $\Pr[\tilde{c}_v \neq c_v] \leq 2\delta$. ■

By the result above, we get that the Hadamard code is a good 2-query LCC that maps k-bit messages into 2^k-bit codewords. That seems wasteful, but it turns out that this is pretty much optimal: every 2-query binary LCC must map k-bit messages into $2^{\Omega(k)}$-bit codewords [KdW03].

5.2 Reed–Muller codes are great LCCs

Above we saw that Hadamard codes are essentially optimal 2-query LCCs. But Hadamard codes are a special case of Reed–Muller codes, so it makes sense to study the LCC properties of more general Reed–Muller codes too.

Theorem 6 *Consider a Reed–Muller code* $\mathsf{RM}_{q,m,r}$ *with* $r < q - 1$. *Then,* $\mathsf{RM}_{q,m,r}$ *has dimension* $\binom{m+r}{r}$ *and is an* $(\ell = r + 1, \delta, \delta(r + 1))$*-LCC.*

Proof: We describe only the local decoder and leave its analysis and the claim about the dimension of the Reed–Muller code to the problem set.

We see vectors of $\mathbb{F}_q^{n=q^m}$ as being indexed by vectors $v \in \mathbb{F}_q^m$. Given $y \in \mathbb{F}_q^n$ such that $d(y, c) < \delta n$ for some $c \in \mathsf{RM}_{q,m,r}$ and a coordinate $v \in \mathbb{F}_{q^m}$, the decoder proceeds as follows:

1. Sample z uniformly at random from $\mathbb{F}_q^m$ and consider the line $L = \{v + \lambda z : \lambda \in \mathbb{F}_q\}$.
2. Let $S = \{\lambda_1, \ldots, \lambda_{r+1}\} \subseteq \mathbb{F}_q^*$ be an arbitrary subset of size $r + 1$. Query $y_{v+\lambda_i z}$ for all $\lambda_i \in S$.
3. Using polynomial interpolation, recover the unique polynomial $Q \in \mathbb{F}_q[\lambda]$ of degree at most r such that $Q(\lambda_i) = y_{v+\lambda_i z}$ for all $i \in [r + 1]$.
4. Output $Q(0)$ as the guess for c_v.

You will also see a more sophisticated local decoder in the problem set. ■

In particular, setting $r = 2$ and $q = 4$ we get a linear 3-query LCC over $\mathbb{F}_4$ that maps messages of length $k = \Theta(m^2)$ to codewords of length $q^m = 2^{O(\sqrt{k})}$, with error $\varepsilon < 1/2$ whenever $\delta < 1/6$. It turns out that this is essentially optimal, at least for linear LCCs.

References

[ASSY23] Emmanuel Abbe, Ori Sberlo, Amir Shpilka, and Min Ye. Reed-Muller codes. *Foundations and Trends in Communications and Information Theory*, 20(1-2):1–156, 01 2023.

[BR60] R. C. Bose and Dwijendra K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Inf. Control.*, 3(1):68–79, 1960.

[CU57] L. Carlitz and S. Uchiyama. Bounds for exponential sums. *Duke Mathematical Journal*, 24(1):37 – 41, 1957.

[Hoc59] Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2:147–156, 1959.

[KdW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 106–115, New York, NY, USA, 2003. Association for Computing Machinery.

[KKM+16] Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D. Pfister, Eren Şaşoğlu, and Rüdiger Urbanke. Reed-Muller codes achieve capacity on erasure channels. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 658–669, New York, NY, USA, 2016. Association for Computing Machinery.

[Mul54] David E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the IRE Professional Group on Electronic Computers*, EC-3(3):6–12, 1954.

[Ree54] Irving Reed. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, 1954.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Found. Trends Theor. Comput. Sci.*, 6(3):139–255, October 2012.