Reed–Muller and Kerdock Codes

In ▶ Chapter 6, we studied Reed–Solomon codes, codes whose codewords are the evaluation of polynomials in one variable of degree at most $k-1$ at the elements of $\mathbb{F}_q \cup \{\infty\}$. Reed–Solomon codes are short length codes, where the length n is bounded by $q+1$, and only useful when we take the field to be large. The alternant codes which we constructed from generalised Reed–Solomon codes in ▶ Chapter 7 allowed us to construct codes over small fields and we put this to good use. In this chapter we will consider another generalisation of Reed–Solomon codes, codes whose codewords are the evaluation of polynomials in many variables. This again allows us to construct linear codes over small fields and we will restrict our attention, for the most part, to binary linear codes. It will turn out that these codes are not asymptotically good. Nevertheless, they are an important class of codes which are widely implemented due to the availability of fast decoding algorithms. One example of such a decoding algorithm is the majority-logic decoding algorithm that we will study here. We will then go on and construct Kerdock codes which are certain subcodes of the second-order Reed–Muller codes. These codes can give examples of non-linear codes with parameters for which no linear code exists.

9.1 Binary Reed–Muller Codes

A **Boolean function** from $\mathbb{F}_2^m$ to $\mathbb{F}_2$ is the evaluation map of a polynomial with coefficients from $\mathbb{F}_2$ in m variables generated by monomials in which the degree of any particular indeterminate is at most 1.

Note that for both elements x of $\mathbb{F}_2$, $x^2 = x$, so the function defined by the evaluation of the polynomial $x_1^2 x_2^3 x_3$ at the elements of $\mathbb{F}_2^3$ and the polynomial $x_1 x_2 x_3$ will be the same. Therefore, it makes sense that when considering evaluations of polynomials in many variables over $\mathbb{F}_2$, we restrict our attention to Boolean functions.

The **r-th order Reed–Muller code** is a binary code $R(r, m)$ of length 2^m defined by

$$R(r, m) = \{(f(a_1), \ldots, f(a_{2^m})) \mid \deg f \leqslant r\},$$

S. Ball, *A Course in Algebraic Error-Correcting Codes*, Compact Textbooks in Mathematics,
https://doi.org/10.1007/978-3-030-41153-4_9

where $\{a_1, \ldots, a_{2^m}\}$ is the set of vectors of $\mathbb{F}_2^m$ and f runs through all Boolean functions that are defined by polynomials in m indeterminates of degree at most r.

The code $R(r, m)$ is a linear code over $\mathbb{F}_2$, since

$$(f(a_1), \ldots, f(a_{2^m})) + (g(a_1), \ldots, g(a_{2^m})) = ((f+g)(a_1), \ldots, (f+g)(a_{2^m})).$$

The vector space of Boolean functions of degree at most r in m variables has a canonical basis, which is the set of monomials of degree at most r in m variables and degree at most one in any particular variable. Therefore, the code $R(r, m)$ has a generator matrix whose rows are indexed by these monomials. For example, the set of monomials

$$\{1, x_1, \ldots, x_m, x_1x_2, \ldots, x_{m-1}x_m\}$$

is a basis for the vector space of Boolean functions in m variables of degree at most 2.

Example 9.1

The 11×16 matrix

$$\mathrm{G} = \begin{pmatrix} 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\ 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \end{pmatrix} \begin{matrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_1x_2 \\ x_1x_3 \\ x_1x_4 \\ x_2x_3 \\ x_2x_4 \\ x_3x_4 \end{matrix}$$

is a generator matrix of the code $R(2, 4)$, with the rows being indexed by the monomials in four variables of degree at most two. ■

We have already proved the following lemma.

Lemma 9.2 *$R(r, m)$ is a linear code of length 2^m and of dimension*

$$1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}.$$

Since $R(r, m)$ is linear, Lemma 4.1 implies that its minimum distance is equal to the minimum weight of a non-zero codeword. In the above example, evidently $R(2, 4)$ has codewords of weight 4 and this is indeed its minimum distance. In Theorem 9.3 we will

calculate the minimum distance for binary Reed–Muller codes. Later, in Theorem 9.15, we will calculate the minimum distance for non-binary Reed–Muller codes.

Theorem 9.3
The minimum distance of $R(r, m)$ is 2^{m-r}.

Proof
By induction on m. If $m = r$, then the evaluation of the polynomial $X_1 \cdots X_r$ is a codeword of weight one.

Suppose that the minimum distance of $R(r, m)$ is 2^{m-r}.

Order the vectors of $\mathbb{F}_2^{m+1}$ so that the first 2^m vectors have $x_{m+1} = 0$.

A codeword $(u, u + v)$ of $R(r, m + 1)$ is the evaluation of a polynomial

$$f(X) + X_{m+1}g(X),$$

where $f(X)$ is a polynomial of degree at most r in m variables and $g(X)$ is a polynomial of degree at most $r - 1$ in m variables. Then $u \in R(r, m)$, since it is the evaluation of $f(X)$ and $v \in R(r - 1, m)$, since it is the evaluation of $g(X)$.

If $u = 0$, then the codeword is $(0, v)$ and by induction has non-zero weight at least $2^{m-(r-1)}$.

If $u + v = 0$, then $u = v \in R(r - 1, m)$ and so the codeword $(u, 0)$ has non-zero weight at least $2^{m-(r-1)}$.

If neither u nor $u + v$ is zero, then $(u, u + v)$ has weight at least $2 \cdot 2^{m-r} = 2^{m-r+1}$, since both u and $u + v$ are in $R(r, m)$.

Thus, the minimum weight of a non-zero codeword of $R(r, m + 1)$ is 2^{m-r+1}. By Lemma 4.1, the minimum weight of a non-zero codeword of a linear code is equal to its minimum distance. □

9.2 Decoding Reed–Muller Codes

The popularity of Reed–Muller codes in real-world applications is due in part to the fact that there are fast decoding algorithms, the most common of which is the focus of this section. Before we consider this decoding algorithm, we first prove a couple of lemmas which prove some properties of Boolean functions.

For each non-empty subset J of $\{1, \ldots, m\}$, let

$$f_J(X) = \prod_{j \in J} X_j$$

and define

$$f_\emptyset(X) = 1.$$

Then

$$\{f_J(X) \mid J \subseteq \{1, \ldots, m\},\ |J| \leqslant r\}$$

is a basis for the space of polynomials in m variables of degree at most r whose evaluations define Boolean functions.

We will exploit the following lemma repeatedly.

Lemma 9.4 *Let J be a subset of $\{1, \ldots, m\}$. Suppose*

$$g(X) = \sum_{L \subseteq \{1,\ldots,m\}} a_L f_L(X),$$

for some $a_L \in \mathbb{F}_2$, where the sum is over all subsets L of size at most $m - |J|$.

Then

$$\sum_{x \in \mathbb{F}_2^m} f_J(x) g(x) = a_{\{1,\ldots,m\} \setminus J}.$$

Proof

Let $K \subseteq \{1, \ldots, m\}$.

If there is an $i \in \{1, \ldots, m\} \setminus K$, then

$$\sum_{\{x \in \mathbb{F}_2^m \mid x_i = 0\}} f_K(x) = \sum_{\{x \in \mathbb{F}_2^m \mid x_i \neq 0\}} f_K(x).$$

This implies

$$\sum_{x \in \mathbb{F}_2^m} f_K(x) = 0, \tag{9.1}$$

unless $K = \{1, \ldots, m\}$.

Then

$$\sum_{x \in \mathbb{F}_2^m} f_J(x) g(x) = \sum_{L \subseteq \{1,\ldots,m\}} \sum_{x \in \mathbb{F}_2^m} a_L f_J(x) f_L(x),$$

where the first sum on the right-hand side is over all subsets L of size at most $m - |J|$.

This expression is equal to

$$\sum_{L \subseteq \{1,\ldots,m\}} a_L \sum_{x \in \mathbb{F}_2^m} f_{J \cup L}(x) = a_{\{1,\ldots,m\} \setminus J},$$

by (9.1). □

Theorem 9.5
The dual of the code $R(r, m)$ is the Reed–Muller code $R(m - r - 1, m)$.

Proof
A codeword u of $R(r, m)$ is the evaluation of a polynomial

$$g(X) = \sum_{K \subseteq \{1,\ldots,m\}} a_K f_K(X),$$

for some $a_K \in \mathbb{F}_2$, where the sum is over all subsets of size at most r.

A codeword v of $R(m - r - 1, m)$ is the evaluation of

$$h(X) = \sum_{L \subseteq \{1,\ldots,m\}} b_L f_L(X),$$

for some $b_L \in \mathbb{F}_2$, where the sum is over all subsets of size at most $m - r - 1$.

The inner product of u and v is

$$\sum_{x \in \mathbb{F}_2^m} h(x)g(x) = \sum_{x \in \mathbb{F}_2^m} \sum_K \sum_L a_K b_L f_K(x) f_L(x)$$

$$= \sum_K \sum_L a_K b_L \sum_{x \in \mathbb{F}_2^m} f_{K \cup L}(x) = 0,$$

by Lemma 9.4.

Therefore,

$$R(m - r - 1, m) \subseteq R(r, m)^\perp.$$

By Theorem 9.3, the sum of the dimensions of $R(r, m)$ and $R(m - r - 1, m)$ is 2^m, which is the length of the codes.

Hence,

$$\dim R(m - r - 1, m) = \dim R(r, m)^\perp.$$

□

The following lemma is fundamental to the decoding algorithm.

Lemma 9.6 *Let*

$$g(X) = \sum_{K \subseteq \{1,\ldots,m\},\ |K| \leqslant r} b_K f_K(X),$$

where $b_K \in \mathbb{F}_2$ and let J be a subset of $\{1, \ldots, m\}$ of size r.

For all 2^{m-r} choices of $a_i \in \mathbb{F}_2$, $i \in \{1, \ldots, m\} \setminus J$,

$$\sum_{x \in \mathbb{F}_2^n} g(x) \prod_{i \in \{1,\ldots,m\} \setminus J} (x_i + a_i) = b_J.$$

Proof

When we expand the product in the sum, all terms have degree less than m except those coming from

$$g(x) \prod_{i \in \{1,\ldots,m\} \setminus J} x_i = g(x) f_{\{1,\ldots,m\} \setminus J}(x).$$

The lemma follows from Lemma 9.4. □

We are now in a position to describe a decoding algorithm for Reed–Muller codes, which is an example of a **majority-logic decoding** algorithm. Let v be the received vector, whose coordinates v_x are indexed by the vectors $x \in \mathbb{F}_2^m$. For each subset J of $\{1, \ldots, m\}$ of size r, we perform a test. We wish to determine whether u_J is zero or one, where the sent codeword u is the evaluation of

$$\sum_{J \subseteq \{1,\ldots,m\},\ |J| \leqslant r} u_J f_J(X).$$

For all 2^{m-r} choices of $a_i \in \mathbb{F}_2$, $i \in \{1, \ldots, m\} \setminus J$, we calculate

$$\sum_{x \in \mathbb{F}_2^m} v_x \prod_{i \in \{1,\ldots,m\} \setminus J} (x_i + a_i).$$

If the result of this test is 1 in the majority of cases, then we conclude that $u_J = 1$ and vice versa, if it is 0 in the majority of cases, then we conclude that $u_J = 0$. Once we have completed this for all subsets J of $\{1, \ldots, m\}$ of size r, we subtract the evaluation of

$$\sum_{K \subseteq \{1,\ldots,m\},\ |K| = r} u_K f_K(X),$$

from the received vector and continue with the subsets of size $r - 1$ supposing that, if we are correctly decoding, we now have a corrupted codeword of $R(r - 1, m)$.

All that remains to be shown, to prove that this decoding algorithm will correct up to $2^{m-r-1} - 1$ error bits, is to show that an error bit will only affect one of the tests. Before we prove this in Lemma 9.8, we consider an example.

Example 9.7

Suppose that we have encoded using $R(2, 4)$ and have received

$$v = (1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0),$$

where the vectors of $\mathbb{F}_2^4$ are ordered as in the matrix G in Example 9.1.

We calculate

$$w = v\mathrm{G}^t = (1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0).$$

The coordinates are indexed by subsets of $\{1, 2, 3, 4\}$ of size at most 2, as in Example 9.1. Indexing the coordinates explicitly

J	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{12\}$	$\{13\}$	$\{14\}$	$\{23\}$	$\{24\}$	$\{34\}$
w_J	1	0	1	1	0	1	1	0	0	1	0

where

$$w_J = \sum_{x \in \mathbb{F}_2^m} v_x f_J(x)$$

are the coordinates of w.

We start by determining u_J for the subsets J of size $r = 2$.

To determine $u_{\{12\}}$, we make $2^{m-r} = 4$ tests by calculating

$$\sum_{x \in \mathbb{F}_2^m} v_x x_3 x_4, \sum_{x \in \mathbb{F}_2^m} v_x (x_3 x_4 + x_3), \sum_{x \in \mathbb{F}_2^m} v_x (x_3 x_4 + x_4)$$

and

$$\sum_{x \in \mathbb{F}_2^m} v_x (x_3 x_4 + x_3 + x_4 + 1),$$

which is

$$w_{\{34\}}, w_{\{34\}} + w_{\{3\}}, w_{\{34\}} + w_{\{4\}} \text{ and } w_{\{34\}} + w_{\{3\}} + w_{\{4\}} + w_{\emptyset},$$

respectively.

The results of these tests are 0, 1, 0, 0, respectively, so we decode $u_{\{12\}}$ as 0, since there are a majority of zeros.

The following table lists the results of these tests for all subsets of size 2 and indicates the majority decision.

$u_{\{12\}}$	$0, 1, 0, 0 \to 0$	$u_{\{13\}}$	$1, 0, 1, 1 \to 1$	$u_{\{14\}}$	$0, 1, 1, 1 \to 1$
$u_{\{23\}}$	$0, 0, 0, 1 \to 0$	$u_{\{24\}}$	$1, 1, 0, 1 \to 1$	$u_{\{34\}}$	$1, 1, 0, 1 \to 1$

Based on the results of those tests, we subtract

$$(0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1)\mathrm{G}$$

from v and get

$$v^1 = v + (0,0,0,0,0,0,1,1,0,1,1)\mathrm{G} = (1,1,1,1,0,0,1,0,1,1,1,1,0,0,0,0).$$

If we are decoding correctly, v^1 should be a (possibly) corrupted codeword of $R(1,4)$. To determine u_J, where J is a subset of size 1, we repeat the above.

We calculate

$$w^1 = v^1 \mathrm{G}_1^t,$$

where G_1 is the generator matrix of $R(3,m)$. This vector will have coordinates

$$w_K^1 = \sum_{x \in \mathbb{F}_2^m} f_K(x) v_x^1,$$

where K is a subset of $\{1,2,3,4\}$ of size at most 3.

Indexing the coordinates explicitly as before

K	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{12\}$	$\{13\}$	$\{14\}$
w_K^1	1	0	1	1	0	0	0	0

K	$\{23\}$	$\{24\}$	$\{34\}$	$\{123\}$	$\{124\}$	$\{134\}$	$\{234\}$
w_K^1	1	0	0	0	0	1	0

allows us to perform $2^{m-(r-1)} = 8$ tests for each u_J.

To determine $u_{\{1\}}$, we make 8 tests by calculating

$$w^1_{\{234\}},\ w^1_{\{234\}} + w^1_{\{23\}},\ w^1_{\{234\}} + w^1_{\{24\}},\ w^1_{\{234\}} + w^1_{\{34\}},\ w^1_{\{234\}} + w^1_{\{23\}} + w^1_{\{24\}} + w^1_{\{2\}},$$

$$w^1_{\{234\}} + w^1_{\{23\}} + w^1_{\{34\}} + w^1_{\{3\}},\ w^1_{\{234\}} + w^1_{\{24\}} + w^1_{\{34\}} + w^1_{\{4\}}$$

and

$$w^1_{\{234\}} + w^1_{\{23\}} + w^1_{\{24\}} + w^1_{\{34\}} + w^1_{\{2\}} + w^1_{\{3\}} + w^1_{\{4\}} + w^1_{\emptyset}.$$

The results of these tests are

$u_{\{1\}}$	$0,1,0,0,0,0,0,0 \to 0$	$u_{\{2\}}$	$1,1,1,1,1,0,1,1 \to 1$
$u_{\{3\}}$	$0,0,0,0,0,1,0,0 \to 0$	$u_{\{4\}}$	$0,0,0,1,0,0,0,0 \to 0$

Based on the results of the tests, we subtract

$$(0,0,1,0,0,0,0,0,0,0)\mathrm{G}$$

from v_1 and get

$$v_2 = v + (0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1)\mathrm{G} = (1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1).$$

Summing

$$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)\mathrm{G}$$

to v_2 we have that

$$v + (1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1)\mathrm{G} = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0).$$

Therefore, we have determined that the error is in the 7-th bit, that the uncoded string

$$u = (1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1)$$

and that the sent codeword was $u\mathrm{G}$. ■

To finish this section, we prove that an error bit affects exactly one of the tests when testing a corrupted codeword of $R(r, m)$. Since we perform 2^{m-r} tests, this implies that we can correct up to $2^{m-r-1} - 1$ error bits, or in other terms $\frac{1}{2}d - 1$ error bits, since the minimum distance d of $R(r, m)$ is 2^{m-r}.

Lemma 9.8 *Suppose that e is a vector of $\mathbb{F}_2^{2^m}$ of weight one, whose coordinates are indexed by the vectors of $\mathbb{F}_2^m$. Let J be a subset of $\{1, \ldots, m\}$. For all but one of the choices of a, whose coordinates $a_i \in \mathbb{F}_2$ for $i \in \{1, \ldots, m\} \setminus J$,*

$$\sum_{x \in \mathbb{F}_2^m} e_x \prod_{i \in \{1,\ldots,m\} \setminus J} (x_i + a_i) = 0,$$

where e_x is the coordinate of e indexed by x.

Proof

Let y be the vector of $\mathbb{F}_2^m$ indexing the coordinate where the vector e has a 1.

The vector e is the evaluation of

$$\prod_{i=1}^{m} (X_i + y_i + 1),$$

since it is zero unless $X_i = y_i$ for all $i = 1, \ldots, m$.

Hence, for all $x \in \mathbb{F}_2^m$,

$$e_x \prod_{i \in \{1,\ldots,m\} \setminus J} (x_i + a_i) = \prod_{i=1}^{m} (x_i + y_i + 1) \prod_{i \in \{1,\ldots,m\} \setminus J} (x_i + a_i),$$

which will contain a factor $x_i^2 + x_i$ (and is therefore zero) unless $a_i = y_i + 1$ for all $i \in \{1, \ldots, m\} \setminus J$. □

To decode using this majority-logic decoding algorithm we perform at most 2^m tests k times, where k is the dimension of the code. This is less than n^2 tests, where n is the length of the code. Each test involves summing less than n terms, so the decoding algorithm is completed in a number of steps which is polynomial in the length of the code. This should be compared to syndrome decoding from ▶ Chapter 4, which involved searching through a look-up table with a number of entries which is exponential in n. For this reason Reed–Muller codes and the majority-logic decoding algorithm are widely implemented. However, they do not give a sequence of asymptotically good codes. Although the relative minimum distance is 2^{-r}, which we can bound away from zero by fixing r, the transmission rate of $R(r, m)$ is less than

$$\frac{r}{n}\binom{\log n}{r}$$

which tends to zero as n tends to infinity.

9.3 Kerdock Codes

A codeword of $R(2, m) \setminus R(1, m)$ is the evaluation of polynomials of the form

$$q(X) + \ell(X) \text{ or } q(X) + \ell(X) + 1,$$

where $\ell(X)$ is a linear form in m variables and

$$q(X) = \sum_{1 \leqslant i < j \leqslant m} a_{ij} X_i X_j$$

is a non-zero quadratic form.

If the quadratic form $q(X)$ has maximum rank, then we will prove that, for all the linear forms $\ell(X)$, these codewords will have large weight. Therefore, if we can find a set of quadratic forms whose differences are quadratic forms of maximum rank, then the distance between any two codewords will be large. In this section we will develop and formalise this idea.

Let $\mathrm{A} = (a_{ij})$ be the symmetric matrix defined by the symmetric bilinear form

$$b(X, Y) = q(X + Y) - q(X) - q(Y) = \sum_{1 \leqslant i < j \leqslant m} a_{ij}(X_i Y_j + X_j Y_i) = X^t \mathrm{A} Y.$$

The **rank** of the bilinear form $b(X, Y)$ is defined to be the rank of A.

Lemma 9.9 *Suppose m is even. The evaluation of*

$$\sum_{i=1}^{m/2} X_{2i-1}X_{2i}$$

at the vectors of $\mathbb{F}_2^m$ has $2^{m-1}+2^{m/2-1}$ zeros.

Proof

There are $2^{m/2}$ zeros of the form $(0, x_2, 0, x_4, \ldots, 0, x_m)$.

If $x_{2i-1} \neq 0$ for some $i = 1, \ldots, m/2$, then one of the x_{2i} is determined by

$$\sum_{i=1}^{m/2} x_{2i-1}X_{2i} = 0,$$

which gives $2^{m/2-1}(2^{m/2}-1)$ zeros of this form, $2^{m/2-1}$ zeros for each non-zero vector $(x_1, x_3, \ldots, x_{m-1})$.

Hence, there are precisely $2^{m-1}+2^{m/2-1}$ zeros when evaluated at the vectors of $\mathbb{F}_2^m$. □

We are going to construct codes whose codewords are the evaluation of the sum of a quadratic form and a linear form. For this reason, we want to know the weights of the vectors which are the evaluations of these Boolean functions.

Lemma 9.10 *Suppose m is even, $q(X)$ is a quadratic form and $\ell(X)$ is a linear form. If the bilinear form associated to $q(X)$ has rank m, then the evaluation of $q(X)+\ell(X)$ at the vectors of $\mathbb{F}_2^m$ has either $2^{m-1}+2^{m/2-1}$ or $2^{m-1}-2^{m/2-1}$ zeros.*

Proof

Dickson's theorem, Exercise 9.2, implies that there is a basis of $\mathbb{F}_2^m$ with respect to which $q(x)+\ell(X)$ is

$$\sum_{i=1}^{m/2}(X_{2i-1}X_{2i} + a_{2i-1}X_{2i-1} + a_{2i}X_{2i}).$$

This is equal to

$$\sum_{i=1}^{m/2}(X_{2i-1}+a_{2i})(X_{2i}+a_{2i-1}) + b$$

for some $b \in \mathbb{F}_2$. By Lemma 9.9, the evaluation of $q(X)+\ell(X)$ has either $2^{m-1}+2^{m/2-1}$ zeros or $2^m-(2^{m-1}+2^{m/2-1})$ zeros, depending on whether $b=0$ or 1. □

Let K be a set of symmetric $m \times m$ matrices over $\mathbb{F}_2$, which have zeros on the diagonal, and which have the property that the matrix $\mathrm{A} - \mathrm{A}'$ has rank m for all distinct $\mathrm{A}, \mathrm{A}' \in K$.

No two matrices in K can have the same first row, since their difference is of rank m. The entries on the diagonal of the matrices in K are zero, so the top-left entry of a matrix in K is zero. Hence, we have that

$$|K| \leqslant 2^{m-1}.$$

For each $\mathrm{A} = (a_{ij}) \in K$, let

$$q_{\mathrm{A}}(X) = \sum_{1 \leqslant i < j \leqslant m} a_{ij} X_i X_j.$$

Let $C(K)$ be the code whose codewords are the evaluation at the vectors of $\mathbb{F}_2^m$ of

$$q_{\mathrm{A}}(X) + \ell(X) \quad \text{or} \quad q_{\mathrm{A}}(X) + \ell(X) + 1,$$

for all $A \in K$ and for all linear forms $\ell(X)$.

Theorem 9.11

Suppose that m is even. The code $C(K)$ is a binary block code of length 2^m, size $|K||R(1, m)|$ and minimum distance $2^{m-1} - 2^{m/2-1}$.

Proof

The distance between the evaluation of

$$q_{\mathrm{A}}(X) + \ell(X) + b$$

and

$$q_{\mathrm{A}'}(X) + \ell'(X) + b'$$

is the weight of the evaluation of

$$q_{\mathrm{A}-\mathrm{A}'}(X) + \ell(X) - \ell'(X) + b - b'.$$

Since, $\mathrm{A} - \mathrm{A}'$ has rank m, Lemma 9.10 implies that this distance is at least $2^{m-1} - 2^{m/2-1}$. □

A **Kerdock code** is a code $C(K)$ where $|K| = 2^{m-1}$. Thus, for a Kerdock code, K is of maximum size and the set K is called a **Kerdock set**. A Kerdock code is a binary

block code of length 2^m, it has minimum distance $2^{m-1} - 2^{m/2-1}$ and size 2^{2m}, i.e. it is a $(2^m, 2^{2m}, 2^{m-1} - 2^{m/2-1})_2$ code.

There are many non-equivalent Kerdock codes. Indeed, if $m - 1$ is not prime, then there are at least $2^{\sqrt{m}/2}$ inequivalent Kerdock codes of length 2^m. However, a sequence of Kerdock codes, whose lengths tend to infinity, is asymptotically bad. Although the relative minimum distance tends to $\frac{1}{2}$, the transmission rate is $2m/2^m$, which tends to zero.

Kerdock codes are of interest because they can be non-linear. The algebraic and geometric nature of their construction allows for non-trivial decoding algorithms to be implemented. The fact that Kerdock codes can be non-linear opens up the possibility of constructing codes with parameter sets for which linear codes do not exist.

Example 9.12
Consider the set of 4×4 matrices over $\mathbb{F}_2$

$$\left\{ \begin{pmatrix} 0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix}, \begin{pmatrix} 0&0&1&0\\0&0&0&1\\1&0&0&1\\0&1&1&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&1\\0&0&1&1\\0&1&0&1\\1&1&1&0 \end{pmatrix} \right\}.$$

This set of matrices can be extended to a set K of 8 matrices with the property that the difference of any two matrices has rank 4, see Exercise 9.5. Thus, K is a Kerdock set and, by Theorem 9.11, the binary Kerdock code $C(K)$ is a $(16, 256, 6)_2$ code. We proved in Example 4.20 that there is no binary linear code with these parameters. This code is the **Nordstrom–Robinson** code. ■

9.4 Non-binary Reed–Muller Codes

Until now we have only considered Reed–Muller codes over $\mathbb{F}_2$, but one can naturally generalise the definition of a Reed–Muller code over a general finite field $\mathbb{F}_q$. The codewords of $R_q(r, m)$ are the evaluations at the vectors of $\mathbb{F}_q^m$ of polynomials of degree at most r in m variables, where the degree in any particular variable is at most $q - 1$. The number of vectors in an m-dimensional vector space over $\mathbb{F}_q$ is q^m, so the length of the linear code $R_q(r, m)$ is q^m. Its dimension is more difficult to calculate, see Exercise 9.7. In the following examples, we calculate the dimension for some specific cases and some low weight codewords, which we will then go on and prove are of minimum non-zero weight.

Example 9.13
Suppose $r \leqslant q - 1$. The evaluation of any polynomial in m variables of degree at most r will be a codeword of $R_q(r, m)$. The set

$$\{X_1^{c_1} \cdots X_m^{c_m} \mid c_1 + \cdots + c_m \leqslant r\}$$

is a basis for the space of polynomials in m variables of degree at most r. Hence, the dimension of $R_q(r, m)$ is

$$\binom{m+r}{r}$$

since this is the number of non-negative integer solutions to

$$c_1 + \cdots + c_m \leqslant r.$$

Let $g(X_1)$ be a polynomial of degree r with r distinct roots in $\mathbb{F}_q$. The evaluation of g is a codeword with precisely rq^{m-1} zero coordinates; a zero coordinate being indexed by a vector of $\mathbb{F}_q^m$ whose first coordinate is a root of g. Therefore, $R_q(r, m)$ has codewords of weight $(q - r)q^{m-1}$. ■

Example 9.14

The space of polynomials of degree at most 3 in three variables in which no variable has an exponent larger than 2 has a basis

$$\{1, X_1, X_2, X_3, X_1^2, X_2^2, X_3^2, X_1X_2, X_1X_3, X_2X_3,$$

$$X_1^2X_2, X_1^2X_3, X_2^2X_1, X_2^2X_3, X_3^2X_1, X_3^2X_2, X_1X_2X_3\}.$$

Therefore, the code $R_3(3, 3)$ is a 17-dimensional ternary linear code of length 27. We could also have arrived at this conclusion by considering a monomial basis for all polynomials of degree at most three in three variables and deleting X_1^3, X_2^3 and X_3^3, see Exercise 9.7. ■

Suppose that $r = a(q - 1) + b$, where $0 \leqslant b \leqslant q - 2$. If $g(X_1)$ is a polynomial of degree b with b distinct roots in $\mathbb{F}_q$, then the evaluation of

$$g(X_1)(X_2^{q-1} - 1) \cdots (X_{a+1}^{q-1} - 1),$$

a polynomial of degree r, is non-zero only when evaluated at

$$x = (x_1, 0, \ldots, 0, x_{a+2}, \ldots, x_m)$$

for some x_1 which is not a root of g. Therefore, $R_q(r, m)$ has a codeword of weight

$$(q - b)q^{m-a-1}.$$

We shall prove that this is the minimum weight of a non-zero codeword in the following theorem, the proof of which is an example of a proof using the polynomial method. This type of proof, which one sees often in combinatorics, attempts to obtain bounds from the fact that the number of zeros of a non-zero polynomial is bounded. The application of the method is often something like the following. Given

a combinatorial object, a polynomial is constructed in such a way that the properties of the combinatorial object are translated into algebraic properties of the polynomial. Usually we are interested in the zeros of the polynomial, often restricted to subsets of a vector space. Here, the polynomial is directly given as the polynomial whose evaluation is the codeword. By bounding from above the number of zeros of the polynomial, we will bound from below the weight of the codeword.

Theorem 9.15
The minimum distance of $R_q(r, m)$ *is* $(q - b)q^{m-a-1}$, *where* $r = a(q - 1) + b$ *and* $0 \leqslant b \leqslant q - 2$.

Proof
By induction on m.

If $m = 1$, then the codewords are the evaluation of a polynomial of degree $r \leqslant q - 1$ in one variable. The polynomial has at most r zeros, so the codeword has weight at least $q - r$. Observe that if $r = q - 1$, then $a = 1$ and $b = 0$ and

$$(q - b)q^{m-a-1} = q/q = 1 = q - r.$$

Suppose that the codeword $u \in R_q(r, m)$ is the evaluation of the polynomial

$$f(X) = f(X_1, \ldots, X_m).$$

We write $f(X)$ as a polynomial in X_m, whose coefficients are polynomials in $X_1, \ldots, X_{m-1}$. Thus,

$$f(X) = \sum_{i=0}^{c} f_i(X_1, \ldots, X_{m-1}) X_m^i,$$

where c is the degree of $f(X)$ in the indeterminate X_m. Note that $f_c(X_1, \ldots, X_{m-1}) \not\equiv 0$ and

$$\deg f_c \leqslant \deg f - c \leqslant r - c.$$

The codeword of $R_q(r - c, m - 1)$ which is the evaluation of f_c has, by induction, at least

$$(q - b')q^{m-a'-2}$$

non-zero coordinates, where $r - c = a'(q - 1) + b'$ and $0 \leqslant b' \leqslant q - 2$.

For any $(x_1, \ldots, x_{m-1})$ such that $f_c(x_1, \ldots, x_{m-1}) \neq 0$, there are at least $q - c$ elements of $\mathbb{F}_q$ for which $f(x_1, \ldots, x_{m-1}, X_m)$ is not zero. Hence, the codeword u has weight at least

$$(q - c)(q - b')q^{m-a'-2}.$$

It remains to prove that

$$(q-c)(q-b')q^{m-a'-2} \geqslant (q-b)q^{m-a-1}.$$

The theorem then follows since, by Lemma 4.1, the minimum distance of a linear code is equal to the minimum weight of a non-zero codeword.

If $a' \leqslant a-2$, then this is clear, so we can assume $a' = a-1$ or a.

Suppose $a' = a-1$ and $(q-c)(q-b') < q-b$. This inequality implies $b' > b$. We have $r = a(q-1)+b$ and $r-c = a'(q-1)+b'$, so

$$c = (a-a')(q-1)+b-b' = q-1+b-b'.$$

Then $(q-c)(q-b') < (q-b)$ implies $(b'-b+1)(q-b') < q-b$, a contradiction.

Suppose $a' = a$ and $(q-c)(q-b') < q(q-b)$. We have $r = a(q-1)+b$ and $r-c = a'(q-1)+b'$, so $c = b-b'$. Then $(q-c)(q-b') < q(q-b)$ implies $(q-b+b')(q-b') < q(q-b)$ which implies $b < b'$ and $c < 0$, a contradiction.

□

9.5 Comments

Reed–Muller codes were introduced by Reed [59] and Muller [53] in the 1950s.

We have taken an algebraic rather than a geometric approach to the majority-logic decoding algorithm. For a geometric description of the algorithm, see Van Lint [74] or MacWilliams and Sloane [50].

Dickson's classification of quadratic form over fields of even characteristic is from [22].

If m is odd, then there are examples of sets K for which Exercise 9.6 is a $(\frac{1}{2}(m^2+m)+1-rm)$-dimensional binary linear code. The 11-dimensional codes ($m=5$ and $r=2$) are the codes which caused a dispute between Apple and Samsung, referred to in James Davis' lecture [20]. They can be found in Corollary 17 ($m=5$, $d=t=2$) on page 455 of MacWilliams and Sloane [50].

Kerdock codes were first considered by Kerdock in [44] in 1972. That there are an exponential number of inequivalent Kerdock codes is proven by Kantor in [41]. Kantor takes a geometric approach to Kerdock codes in the articles [42], a treatment of which can be found in Chapter 12 of Cameron and van Lint's book [17]. The Nordstrom–Robinson code is from [54]. Kerdock codes have applications to quantum mechanics, see [15] and [18].

The non-binary Reed–Muller codes were defined by various authors. Theorem 9.15 is attributed to Kasami, Lin and Peterson [43] in Bishnoi [11], where the proof given here is adapted from.

9.6 Exercises

9.1 *Suppose that we have sent a codeword of the code* $R(2, 4)$*, the coordinates ordered as in Example 9.1, and have received the vector*

$$(0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1).$$

i. Decode the received vector using syndrome decoding.
ii. Decode the received vector using majority-logic decoding.

9.2 *Suppose that* $q(X)$ *is a quadratic form of rank* m *of the type*

$$q(X) = \sum_{1 \leqslant i < j \leqslant m} q_{ij} X_i X_j .$$

Prove that there is a basis of $\mathbb{F}_2^m$*, with respect to which,* $q(X)$ *is*

$$\sum_{i=1}^{m/2} X_{2i-1} X_{2i} .$$

9.3

i. Prove that we can select half the codewords of $R(1, m)$ *so that the* $2^m \times 2^m$ *matrix* H*, whose rows are the selected codewords with zeros changed to minus one, has the property that* $\mathrm{HH}^t = 2^m \mathrm{I}$*, where* I *is the* $2^m \times 2^m$ *identity matrix.*
ii. Prove that for each vector $v \in \mathbb{F}_2^{2^m}$ *there is a codeword* u *of* $R(1, m)$ *such that* $d(u, v) \leqslant 2^{m-1} - 2^{m/2-1}$.

9.4 *Prove that the Kerdock code* $C(K)$ *of length* 2^m *is linear if and only if the Kerdock set* K *is a subspace of the vector space of* $m \times m$ *matrices.*

9.5 *Complete the set of matrices to a Kerdock set* K *of eight matrices and prove that* $C(K)$ *is a non-linear* $(16, 256, 16)$ *code.*

$$\left\{ \begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}, \begin{pmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&1 \\ 0&1&1&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&1 \\ 0&0&1&1 \\ 0&1&0&1 \\ 1&1&1&0 \end{pmatrix} \right\}$$

9.6

i. Prove that the evaluation of

$$q(X) = \sum_{i=1}^{r} X_{2i-1} X_{2i}$$

at the vectors of $\mathbb{F}_2^m$*, has* $2^{m-1} + 2^{m-r-1}$ *zeros.*

ii. Prove that the evaluation of $q(X) + \ell(X)$*, where* $\ell(X)$ *is a linear form and* $q(X)$ *is a quadratic form whose associated bilinear form is of rank* $2r$*, at the vectors of* $\mathbb{F}_2^m$*, has either* $2^{m-1} + 2^{m-r-1}$*,* 2^{m-1} *or* $2^{m-1} - 2^{m-r-1}$ *zeros.*

iii. Suppose that K *is a set of* $m \times m$ *symmetric matrices over* $\mathbb{F}_2$ *with the property that* $\mathrm{A} + \mathrm{A}'$ *has rank at least* $2r$ *for all* $\mathrm{A}, \mathrm{A}' \in K$*. Construct a* $(2^m, |K|2^{m+1}, 2^{m-1} - 2^{m-r-1})_2$ *code.*

iv. Use iii. to construct a $[32, 11, 12]_2$ *code.*

v. Construct a linear code with the same parameters from the code of length 31 *constructed in Exercise 5.6.*

9.7

i. By finding a monomial basis for the space of polynomials in 3 *variables of degree at most* 4*, in which the degree of each variable is at most* 2*, calculate the dimension of* $R_3(4, 3)$.

ii. Prove that if $r \leqslant q - 1$*, then the dimension of* $R_q(r, m)$ *is*

$$\sum_{i=0}^{r} \binom{m+i-1}{m-1}.$$

iii. Prove that the dimension of $R_q(r, m)$ *is*

$$\sum_{i=0}^{r} \sum_{j=0}^{m} (-1)^j \binom{m+i-1-jq}{m-1} \binom{m}{j}.$$