Chapter 98

Evaluation codes

1. Linear evaluation codes

sec-lin-eval

Mathematically we often think of an n-tuple $\mathbf{f} = (f_1, f_2, \ldots, f_j, \ldots, f_n)$ from E^n as listing the values of a function f from the set $\{1, 2, \ldots, j, \ldots, n\}$ to the alphabet E, that function given by $f(j) = f_j$. Or again, we may think of $\mathbf{f}$ as the graph of the function f. A code C in E^n is then a collection of such graphs or evaluation vectors. From this point of view, it is an *evaluation code*.

evaluation code

The index set $\{1, \ldots, j, \ldots, n\}$ is secondary in importance. Any (ordered) n-set

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_j, \ldots, \alpha_n)$$

would have sufficed, and the codewords of C are then the various evaluation vectors

$$\mathbf{f} = \mathbf{ev}_{\boldsymbol{\alpha}}(f) = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_j), \ldots, f(\alpha_n))$$

as f runs through the set of functions associated with the codewords of C.

This approach comes equipped with its own terminology. The large set from which the individual α_j is selected is the *location space* (in the initial example, the positive integers). The particular α_j are then the *location numbers* (playing the role of the integers from 1 to n) that make up $\boldsymbol{\alpha}$, the *location vector* (originally $(1, 2, \ldots, j, \ldots, n)$).

location space
location numbers
location vector

This recasting is particularly useful when the alphabet or *value space* E is a field F. In that case the set $V_{\boldsymbol{\alpha}}$ of all functions

value space

$$f\colon \{\alpha_1, \ldots, \alpha_n\} \longrightarrow F$$

has a natural structure as a vector space over F:

$$h = af + bg \quad \text{is given by} \quad h(\alpha_j) = af(\alpha_j) + bg(\alpha_j)\,, \quad \text{for } a, b \in F\,,\ f, g \in V_{\boldsymbol{\alpha}}\,,$$

the pointwise calculations going on entirely inside the field F.

It is not surprising that $V_{\boldsymbol{\alpha}}$ is an F-space, since its set of evaluation vectors $\mathbf{ev}_{\boldsymbol{\alpha}}(f)$ is exactly the vector space F^n. More pleasing is that $V_{\boldsymbol{\alpha}}$ becomes a ring when provided with the pointwise product

$$c = fg \quad \text{given by} \quad c(\alpha_j) = f(\alpha_j)g(\alpha_j)\,.$$

The space and ring structure can be useful in the study of the codes because of the following lemma.

lem-linear-eval

(98.1). LEMMA. *Let $f, g \in V_{\boldsymbol{\alpha}}$ and $a, b \in F$.*

(a) $\mathbf{ev}_{\boldsymbol{\alpha}}(af + bg) = a\,\mathbf{ev}_{\boldsymbol{\alpha}}(f) + b\mathbf{ev}_{\boldsymbol{\alpha}}(g)$.

(b) $\mathbf{ev}_{\boldsymbol{\alpha}}(fg) = \mathbf{ev}_{\boldsymbol{\alpha}}(f) * \mathbf{ev}_{\boldsymbol{\alpha}}(g)$, *the Hadamard product of vectors.* □

In the special case where the code C comes from a function set W that is a subspace of $V_{\boldsymbol{\alpha}}$, the lemma says that the evaluation code

$$\mathbf{ev}_{\boldsymbol{\alpha}}(W) = \{\, \mathbf{ev}_{\boldsymbol{\alpha}}(w) \mid w \in W \,\}$$

linear evaluation code

is a linear code in F^n; that is, a *linear evaluation code*. The map $w \mapsto \mathbf{ev}_{\boldsymbol{\alpha}}(w)$ then gives a linear transformation from the vector space W to the vector space F^n. When this map is injective, it can be thought of as encoding messages from the space W into the code subspace C of F^n.

2. First order Reed-Muller codes

We introduced the first order Reed-Muller codes in Class 17 (on 17 February 2017) as the duals of the binary extended Hamming codes. Here we recast them as linear evaluation codes (as in the previous section) and as codes associated with the geometry of $\mathbb{F}_2$-vector spaces. We do this both as a warm-up for consideration of all the (generalized) Reed-Muller codes and to give a geometric and more precise version of a theorem proven in Class 17.

Let $A = \mathbb{F}_2^m$. Consider the space W of all affine linear functions h on A given by

$$h\colon (a_1, \dots, a_i, \dots, a_m) \mapsto h_1a_1 + \cdots + h_ia_i + \cdots + h_ma_m + b$$

where $\mathbf{h} = (h_1 \dots, h_i, \dots h_m) \in A$ and $b \in \mathbb{F}_2$. Let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_j, \dots, \alpha_n)$ be an ordering of length $n = 2^m$ for all the elements α_j of the space A. (So $\boldsymbol{\alpha}$ is a "vector of vectors.")

Then A is the location space and $\boldsymbol{\alpha}$ the location vector for the linear evaluation code

$$\mathrm{RM}_{\boldsymbol{\alpha}}(1, m) = \mathbf{ev}_{\boldsymbol{\alpha}}(W) = \{\, \mathbf{ev}_{\boldsymbol{\alpha}}(h) \mid h \in W \,\}$$

The codeword $\mathbf{ev}_{\boldsymbol{\alpha}}(h)$ has

$$h(\alpha_j) = h_1a_1 + \cdots + h_ia_i + \cdots + h_ma_m + b$$

in the position with location number $\alpha_j = (a_1, \dots, a_i, \dots, a_m)$.

Let $\{\, \mathbf{e}^i \mid 1 \le i \le m \,\}$ be the canonical basis for $\mathbb{F}_2^m = A$ so that a typical vector is

$$\beta = (b_1, \dots, b_i, \dots, b_m) = \sum_{i=1}^{m} b_i \mathbf{e}^i \,.$$

On $\mathbb{F}_2^m$ we call the i^{th} coordinate function x_i, so that

$$x_i(\beta) = b_i \,.$$

Then every affine function is a linear combination of the functions x_i and the constant function 1—from above

$$h = \sum_{i=1}^{m} h_i x_i + b1 \,.$$

By Lemma (98.1) we then get

$$\mathbf{ev}_{\boldsymbol{\alpha}}(h) = \sum_{i=1}^{m} h_i \mathbf{ev}_{\boldsymbol{\alpha}}(x_i) + b\mathbf{ev}_{\boldsymbol{\alpha}}(1)\,,$$

and the evaluation code $\mathrm{RM}_{\boldsymbol{\alpha}}(1,m)$ is spanned by (indeed generated by) its linearly independent codewords

$$\mathbf{ev}_{\boldsymbol{\alpha}}(1) \quad \text{and} \quad \mathbf{ev}_{\boldsymbol{\alpha}}(x_i) \quad \text{for } 1 \le i \le m\,.$$

But these are the rows of a generator matrix for a first order Reed-Muller code $\mathrm{RM}(1,m)$, with the column whose location number is β consisting precisely of the (column) vector β with an additional 1 added at its bottom:

$$\begin{aligned} x_m(\beta) &= b_m \\ x_{m-1}(\beta) &= b_{m-1} \\ &\cdots \\ x_i(\beta) &= b_i \\ &\cdots \\ x_1(\beta) &= b_1 \\ 1(\beta) &= 1\,. \end{aligned}$$

For example, consider the case $m = 3$. To make things simpler, let us assume that $\boldsymbol{\alpha}$ is the lexicographic ordering $A = \mathbb{F}_2^3$. (Different orderings produce different but equivalent codes.) the generator matrix of $\mathrm{RM}_{\boldsymbol{\alpha}}(1,3)$ as described is exactly the lexicographic generator matrix XL_3 given in class:

$\boldsymbol{\alpha} =$	000	001	010	011	100	101	110	111
$\mathbf{ev}_\alpha(x_1) =$	0	0	0	0	1	1	1	1
$\mathbf{ev}_\alpha(x_2) =$	0	0	1	1	0	0	1	1
$\mathbf{ev}_\alpha(x_3) =$	0	1	0	1	0	1	0	1
$\mathbf{ev}_\alpha(1) =$	1	1	1	1	1	1	1	1

We arrive at a technical but important result characterizing the first order Reed-Muller codes and their codewords. In many places, some version of the following proposition serves as the definition for a first order Reed-Muller code.

(98.2). PROPOSITION. *The code* $\mathrm{RM}_{\boldsymbol{\alpha}}(1,m)$ *is a first order Reed-Muller code* $\mathrm{RM}(1,m)$*, the dual of an extended binary Hamming code of length* 2^m*.* prop-grm-rm1

It has length 2^m *and dimension* $1+m$ *and consists of the codewords* $\mathbf{ev}_{\boldsymbol{\alpha}}(\mathbf{h},b)$ *for* $\mathbf{h}$ *in the location space* $A = \mathbb{F}_2^m$ *and* $b \in \mathbb{F}_2$ *with the value at location* α *given by*

$$\mathbf{ev}_{\boldsymbol{\alpha}}(\mathbf{h},b)_\alpha = \alpha \cdot \mathbf{h} + b\,.$$

For $\mathbf{h} = \mathbf{0}$ *these are the repetition codewords* $\mathbf{0}$ *and* $\mathbf{1}$*.*

The codewords with $\mathbf{h} \neq \mathbf{0}$ *all have weight* 2^{m-1}*. Specifically,* $\mathbf{ev}_{\boldsymbol{\alpha}}(\mathbf{h},0)$ *has its* 0*'s at the elements of the* $(m-1)$*-subspace* $\mathbf{h}^\perp$ *of* A*, while* $\mathbf{ev}_{\boldsymbol{\alpha}}(\mathbf{h},0)$ *has its* 0*'s at the complement* $A \setminus \mathbf{h}^\perp$*.*

PROOF. With $\alpha = (a_1, \ldots, a_i, \ldots, a_m)$, the function $\alpha \mapsto \alpha\cdot\mathbf{h}+b$ from $\mathbb{F}_2^m$ to $\mathbb{F}_2$ is exactly the function h defined above, hence $\mathbf{ev}_{\boldsymbol{\alpha}}(\mathbf{h},b) = \mathbf{ev}_{\boldsymbol{\alpha}}(h)$. Reordering the vector $\boldsymbol{\alpha}$ results in an equivalent code, and any code equivalent to a Reed-Muller

code is itself a Reed-Muller code. So the first two paragraphs of the proposition follow from the discussion that precedes it.

The last two paragraphs then follow directly and give a more precise description of the weight data provided in a theorem proven in Class 17. □