Chapter 4

Hamming Codes

In the late 1940's Claude Shannon was developing information theory and coding as a mathematical model for communication. At the same time, Richard Hamming, a colleague of Shannon's at Bell Laboratories, found a need for error correction in his work on computers. Parity checking was already being used to detect errors in the calculations of the relay-based computers of the day, and Hamming realized that a more sophisticated pattern of parity checking allowed the correction of single errors along with the detection of double errors. The codes that Hamming devised, the single-error-correcting binary Hamming codes and their single-error-correcting, double-error-detecting extended versions marked the beginning of coding theory. These codes remain important to this day, for theoretical and practical reasons as well as historical.

4.1 Basics

Denote by L_3 the check matrix that we have been using to describe the $[7, 4]$ Hamming code:

$$L_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

It has among its columns each nonzero triple from $\mathbb{F}_2^3$ exactly once. From this and Lemma 3.1.12, we were able to prove that the $[7, 4]$ Hamming code has minimum distance 3. This suggests a general method for building binary Hamming codes. For any r, construct a binary $r \times 2^r - 1$ matrix H such that each nonzero binary r-tuple occurs exactly once as a column of H. Any code with such a check matrix H is a *binary Hamming code* of redundancy r, denoted $\mathrm{Ham}_r(2)$. Thus the $[7, 4]$ code is a Hamming code $\mathrm{Ham}_3(2)$. Each binary Hamming code has minimum weight and distance 3, since as before there are no columns $\mathbf{0}$ and no pair of identical columns. That is, no pair of columns is linearly dependent, while any two columns sum to a third column, giving a triple of linearly dependent columns. Lemma 3.1.12 again applies.

binary Hamming code

As defined, any code that is equivalent to a binary Hamming code is itself a Hamming code, since any permutation of coordinate positions corresponds to a permutation of the columns of the associated check matrix. The new check matrix is still a census of the nonzero r-tuples. Different codes and check matrices may be selected to suit different purposes.

EXAMPLES. The following are check matrices for two $[15, 11]$ binary Hamming codes $\mathrm{Ham}_4(2)$:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The first is the check matrix for a code which has a generator matrix in standard form (see page 35 and Problem 4.1.9 below). The second matrix checks a code which has no generator in standard form, since, for instance, (000000000001111) is a codeword.

The second of the two example check matrices, which we will denote L_4, is the counterpart of the matrix L_3 above, in that its column i contains the binary representation of i. For each positive integer r, let L_r be the $r \times 2^r - 1$ matrix whose i^{th} column is the binary representation of the integer i (with least significant digit at the bottom). Then L_r is the check matrix for a Hamming code $\mathrm{Ham}_r(2)$. We call L_r a *lexicographic check matrix*.

lexicographic check matrix

EXAMPLES. We have given L_3 and L_4 above. We also have the smaller cases

$$L_1 = [1] \text{ and } L_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

which are check matrices for, respectively, the degenerate Hamming code $\mathrm{Ham}_1(2) = \{\mathbf{0}\}$ and $\mathrm{Ham}_2(2)$, the repetition code of length 3.

For a binary Hamming code with lexicographic check matrix L_r, we have an easy version of syndrome decoding available, similar to that for $\mathrm{Ham}_3(2)$ discussed earlier and presented by Shannon under Example 1.3.3. If the vector $\mathbf{x}$ has been received, then to decode we first calculate the syndrome $\mathbf{s} = L_r\mathbf{x}^\top$. If $\mathbf{s}$ is $\mathbf{0}$, then $\mathbf{x}$ is a codeword and no further decoding is required. If $\mathbf{s}$ is not $\mathbf{0}$, then it is the binary representation of some integer, j say, between 1 and $2^r - 1$. We decode by assuming that a single error has occurred in position j of $\mathbf{x}$.

If we add an overall parity check bit to a binary Hamming code $\mathrm{Ham}_r(2)$, then the minimum distance is increased to 4. We then have an *extended Hamming code*, denoted $\mathrm{XHam}_r(2)$. By Problem 2.2.3 this is a 1-error-correcting, 2-error-detecting binary linear $[2^r, 2^r - r]$ code, as originally constructed by Hamming.

extended Hamming code

Begin with the Hamming code $\mathrm{Ham}_r(2)$ given by the lexicographic check matrix L_r and extend by adding an overall parity check bit at the front of each

codeword. The check matrix XL_r for this extended Hamming code $\mathrm{XHam}_r(2)$ is constructed by adding a column r-tuple $\mathbf{0}$ at the beginning of L_r and then adding at the bottom the vector $\mathbf{1}$ composed entirely of 1's.

EXAMPLES.

$$\mathrm{XL}_1 = \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 1 \end{array}\right] \quad \text{and} \quad \mathrm{XL}_2 = \left[\begin{array}{c|ccc} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \end{array}\right]$$

$$\mathrm{XL}_3 = \left[\begin{array}{c|cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

$$\mathrm{XL}_4 = \left[\begin{array}{c|cccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

To see that XL_r is a check matrix for the extended lexicographic Hamming code, first note that its $r+1$ rows are linearly independent rows; so at least it has the right rank. If $\mathbf{c} = (c_1, \ldots, c_n)$ is a word from the Hamming code, then the corresponding extended word is $\mathbf{c}' = (c_0, c_1, \ldots, c_n)$ where $c_0 = \sum_{i=1}^{n} c_i$ is the overall parity check symbol. The codeword $\mathbf{c}'$ has dot product 0 with each of the first r rows, since its original form $\mathbf{c}$ in the Hamming code has dot product 0 with the corresponding row of L_r. Also the dot product of $\mathbf{c}'$ with $\mathbf{1}$ is $c_0 + c_1 + \cdots + c_n = (\sum_{i=1}^{n} c_i) + c_1 + \cdots + c_n = 0$. Therefore XL_r is indeed a check matrix for the extended Hamming code as described.

We can think of our construction of binary Hamming codes as a greedy construction. We begin with an integer r and try to construct the check matrix for a 1-error-correcting binary linear code of redundancy r that, subject to this, is of maximal length. We add new columns to a check matrix, being careful at each stage not to reduce the minimum distance below three. By Lemma 3.1.12 we can do this provided at each stage we add a new column that is not linearly dependent upon any previous column. As our field is $\mathbb{F}_2$, this amounts to avoiding repeat columns and the $\mathbf{0}$ column.

This approach to Hamming codes easily extends to linear codes over finite fields $\mathbb{F}_q$ other than $\mathbb{F}_2$. Again, begin with r and construct a check matrix for a long 1-error-correcting linear code over $\mathbb{F}_q$ with redundancy r. Each time we add a new column to our matrix, we must take care that it does not depend linearly upon any previously chosen column. We avoid $\mathbf{0}$ always, so initially we can choose from among $q^r - 1$ nonzero column r-tuples. When we add in a nonzero column, we remove not just it from further consideration but each of its $q-1$ multiples by the nonzero elements of $\mathbb{F}_q$. Therefore the maximum length possible is $(q^r - 1)/(q-1)$. One easy way of constructing such a matrix of this maximum length is to choose as columns all nonzero r-tuples whose top-most nonzero entry is 1. A linear code over the finite field $\mathbb{F}_q$ is a *Hamming code* of redundancy r, written $\mathrm{Ham}_r(q)$, if it has a check matrix whose collection of

Hamming code

columns contains a unique nonzero scalar multiple of each nonzero r-tuple from $\mathbb{F}_q$. In particular any code that is monomially equivalent to a Hamming code is again a Hamming code. Note that this general definition includes that for binary Hamming codes given above.

EXAMPLES.

(i) Our original description of $\mathrm{Ham}_2(3)$, a ternary Hamming code of redundancy 2, was in terms of the check matrix

$$\begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} .$$

A ternary Hamming code of redundancy 2 can also be constructed from the "lexicographic" check matrix

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix} .$$

(ii) With $q = 9$, $r = 2$, and $n = (9^2 - 1)/(9 - 1) = 10$, one check matrix for a Hamming code $\mathrm{Ham}_2(9)$ of length 10 is

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & i & 1+i & 2+i & 2i & 1+2i & 2+2i \end{bmatrix} ,$$

where i is a square root of $2 = -1$ in $\mathbb{F}_9$.

(4.1.1) THEOREM. *A Hamming code of redundancy $r(\geq 2)$ over the field F, $|F| = q$, is a linear*

$$\left[\frac{q^r - 1}{q - 1}, \frac{q^r - 1}{q - 1} - r, 3\right]$$

code and is a perfect 1-error-correcting code.

PROOF. Set $n = (q^r - 1)/(q - 1)$, the length of the Hamming code C. As the code has redundancy r, its dimension is $n - r$. As discussed above, the argument applying Lemma 3.1.12 to prove that $\mathrm{d}_{\min}(C) = 3$ for binary codes goes over to the general case; so C corrects single errors. The Sphere Packing Condition 2.2.6 for correcting $e = 1$ error then says

$$|C| \cdot |\mathrm{S}_1(*)| \leq |A^n| .$$

Here this takes the form

$$q^{n-r} \cdot \big(1 + (q - 1)n\big) \leq q^n .$$

That is,

$$q^{n-r} \cdot \big(1 + (q - 1)n\big) = q^{n-r} \cdot \big(1 + (q - 1)\frac{q^r - 1}{q - 1}\big) = q^{n-r} \cdot q^r \leq q^n .$$

Thus we in fact have equality in the Sphere Packing Condition 2.2.6 and the Sphere Packing Bound 2.3.6. □

(4.1.2) PROBLEM. *Prove that the family of all q-ary Hamming codes is asymptotically bad, being associated with the point $(0,1)$ of the q-ary code region.*

(4.1.3) PROBLEM. *Prove that for $r \geq 2$ a*

$$\left[\frac{q^r-1}{q-1}, \frac{q^r-1}{q-1} - r, 3\right]$$

linear code over the field $\mathbb{F}_q$ is a Hamming code.

Problem 4.1.3 says that, among linear codes, the Hamming codes are characterized by their parameters — length, dimension, and distance. This is not the case if we drop the assumption of linearity. Of course any coset of a Hamming code $\mathrm{Ham}_r(q)$ has the same parameters as the code. For $q = 2$ and for $r \leq 3$, the converse is true. This is trivial for $r = 1, 2$ and the case $r = 3$ is given as the next problem. For $q = 2$ and $r \geq 4$ nonlinear codes with Hamming parameters exist that are not a coset of any Hamming code. Examples are constructed in Problem 4.1.7 below. For $q > 2$ and $r \geq 2$, nonlinear examples exist as well and can be constructed in a manner related to that of Problem 4.1.7.

(4.1.4) PROBLEM. *Prove that a 1-error-correcting binary code of length 7 that contains 16 vectors must be a coset of a $[7,4]$ Hamming code.*

(4.1.5) PROBLEM. *Let C_1 be an $[n, k_1, d_1]$ linear code over F, and let C_2 be a $[n, k_2, d_2]$ linear code code over F. Form the code*

$$C = \{\ (\mathbf{y}\,;\mathbf{x}+\mathbf{y}) \mid \mathbf{x} \in C_1,\ \mathbf{y} \in C_2\ \}.$$

(a) If C_1 has generator matrix G_1 and C_2 has generator matrix G_2, prove that C is a $[2n, k_1 + k_2]$ linear code over F with generator matrix

$$G = \begin{bmatrix} \mathbf{0} & G_1 \\ G_2 & G_2 \end{bmatrix},$$

where the upper left $\mathbf{0}$ is a $k_1 \times n$ matrix entirely composed of 0's.

(b) Prove that $\mathrm{d}_{\min}(C) = \min(d_1, 2d_2)$.

(c) Under the additional assumption $d_1 > 2d_2$, prove that all codewords of minimum weight in C have the form $(\mathbf{y}\,;\mathbf{y})$, where $\mathbf{y}$ has minimum weight in C_2.

(4.1.6) PROBLEM. *Formulate and prove the appropriate version of Problem 4.1.5 for nonlinear C_1 and C_2.*

(4.1.7) PROBLEM. *In Problem 4.1.5 let $C_1 = \mathrm{Ham}_{r-1}(2)$ and $C_2 = \mathbb{F}_2^n$ where $n = 2^{r-1}-1$. Then C has length $2n = 2^r-2$, dimension $2^{r-1}-1-(r-1)+2^{r-1} = 2^r-r$, and minimum distance 2. Furthermore all codewords of weight 2 in C have the shape $(\mathbf{y}\,;\mathbf{y})$, for $\mathbf{y} \in \mathbb{F}_2^n$ of weight 1. Consider now the code*

$$C^* = \{\ (\mathbf{y}\,;\mathbf{x}+\mathbf{y}\,;c) \mid \mathbf{x} \in \mathrm{Ham}_{r-1}(2),\ \mathbf{y} \in \mathbb{F}_2^n,\ c = \mathbf{y}\cdot\mathbf{1} + f(\mathbf{x})\ \},$$

where $\mathbf{1} \in \mathbb{F}_2^n$ is the vector with all coordinates equal to 1, $\mathbf{y}\cdot\mathbf{1}$ is the usual dot product, and f is any function from $\mathrm{Ham}_{r-1}(2)$ to $\mathbb{F}_2$.

(*a*) *Prove that* C^* *is a code with the same length, size, and minimum distance as* $\mathrm{Ham}_r(2)$.

(*b*) *Prove that* C^* *is linear if and only if the function* f *is a linear map on* $\mathrm{Ham}_{r-1}(2)$ *(that is, for all* $\mathbf{x_1}, \mathbf{x_2} \in \mathrm{Ham}_{r-1}(2)$*, we have* $f(\mathbf{x_1}) + f(\mathbf{x_2}) = f(\mathbf{x_1} + \mathbf{x_2})$*.)*

(*c*) *Prove that, for all* $r \geq 4$*, there exist binary codes* C^* *with the same parameters as the Hamming codes* $\mathrm{Ham}_r(2)$ *that are not equal to a coset of any Hamming code.*

Any check matrix H for a Hamming code $\mathrm{Ham}_r(q)$ can be easily used for syndrome decoding. Upon encountering a nonzero syndrome $\mathbf{s} = H\mathbf{x}^\top$, we must survey the columns of H for a scalar multiple of $\mathbf{s}$. If $\mathbf{s}$ is $\alpha\mathbf{h}$, where $\mathbf{h}$ is the i^{th} column of H, then we assume that the error vector had weight one with entry α in its i^{th} coordinate, and decode accordingly.

(4.1.8) PROBLEM. *Consider the ternary* $[13, 10]$ *Hamming code with check matrix*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{bmatrix} .$$

Decode the received word

$$(2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1) .$$

(4.1.9) PROBLEM. *Consider a binary* $[n, k]$ *linear code* C_r *with redundancy* $r = n - k$ *and standard generator matrix* $\left[I_{k\times k} \,|\, A \right]$*, for some* $k \times r$ *binary matrix* A*.*

(*a*) *Prove that* $\mathrm{d}_{\min}(C) \geq 3$ *if and only if the rows of* A *are pairwise distinct* r*-tuples of weight at least* 2*.*

(*b*) *Using the result from* (*a*)*, prove that the maximum possible number of rows in* A *for a* 1*-error-correcting code* C_r *is* $2^r - 1 - r$ *and that, in this case,* C *is a Hamming code* $\mathrm{Ham}_r(2)$*.*

In constructing codes with a fixed minimum distance d we hope for codes whose dimension is a large fraction of their length or, equivalently, whose redundancy is small compared to their length. Let $s_q(n, f) = \sum_{i=0}^{f} \binom{n}{i}(q-1)^i$ be the volume of a sphere of radius f in $\mathbb{F}_q^n$. The Gilbert-Varshamov Theorem 2.2.7 proved, by a greedy construction of codewords, that there is in $\mathbb{F}_q^n$ a code C with minimum distance d and satisfying $|C| \geq q^n / s_q(n, d-1)$. That is, we can find a code C with distance d and length n whose redundancy $r = n - \log_q(|C|)$ is bounded above by

$$q^r \leq s_q(n, d-1) .$$

In this section we have used greedy constructions instead to construct the check matrices of Hamming codes. We pursue this in Problem 4.1.10 and find, via a greedy construction of a check matrix, that there is a linear code C with minimum distance d and redundancy r whose length n is bounded below by

$$q^r \leq s_q(n, d-2) .$$

This small improvement may be relevant in specific cases; but, since in the limit $(d-1)/n$ and $(d-2)/n$ are indistinguishable, the asymptotic bound of Theorem

2.3.7 is not affected. Although the bounds have a very similar appearance, the two constructions are essentially different in character. The first endeavors to constructs a code of small redundancy with given length, while the second tries to construct a long code with given redundancy.

(4.1.10) PROBLEM. (*a*) *Let H be an $r \times m$ matrix with entries from $\mathbb{F}_q$. For a fixed positive integer d, prove that the number of column r-tuples that linearly depend upon some subset of $d-2$ columns from H is at most $\sum_{i=0}^{d-2} \binom{m}{i}(q-1)^i = s_q(m, d-2)$.*

(*b*) (LINEAR GILBERT-VARSHAMOV BOUND) *Prove that there exists a linear code C over $\mathbb{F}_q$ of minimum distance d and redundancy r whose length n satisfies*

$$q^r \leq s_q(n, d-2) \ .$$

4.2 Hamming codes and data compression

Hamming codes can also be used for data compression allowing a small amount of distortion (loss of information) by "running the machine backwards."

Choose a generator matrix G for a Hamming code $\mathrm{Ham}_r(q)$ of length n over F. Each n-tuple $\mathbf{x}$ from F^n is at distance at most 1 from a unique codeword $\mathbf{c}$ of $\mathrm{Ham}_r(q)$. Instead of storing the n-tuple $\mathbf{x}$, store the smaller message $(n-r)$-tuple $\mathbf{m}$, the unique solution to $\mathbf{m}G = \mathbf{c}$. At decompression, the stored message $\mathbf{m}$ is "encoded" to $\mathbf{c} = \mathbf{m}G$, which differs from the original data vector $\mathbf{x}$ in at most one position. This works because spheres of radius one around the codewords of $\mathrm{Ham}_r(q)$ cover the whole codespace F^n.

Consider a code C of length n over the alphabet A. In general the *covering radius* of a code C, denoted $\mathrm{cr}(C)$, is the smallest number r such that the spheres of radius r around the codewords of C cover the entire codespace A^n, that is, $A^n = \bigcup_{\mathbf{c}\in C} \mathrm{S}_r(\mathbf{c})$. Data compression problems are basically dual to those of error correction. A good code for correcting errors has a large number of words but still has large minimum distance. A good data compression code has a small number of words but still has a small covering radius. As with correction, these are conflicting goals.

covering radius

Data compression questions have been considered for almost as long as those of error correction. Sometimes they have been phrased in terms of the "football pools" or lottery problem. Typically in the lottery, each ticket purchased contains a collection of numbers (selected by the purchaser and from some fixed range A). Whether or not a given ticket is a winner or loser depends upon how well its numbers match those of a master ticket which is selected at a later time. If in order to win it is not necessary for a given ticket to match all of the master lottery numbers but only miss at most f of them, then a question arises. What is the smallest number of lottery tickets I must choose and buy in order to guarantee that I have a ticket on which at most f numbers are wrong? What is being sought is a small code (the collection of purchased lottery tickets) that has covering radius at most f. For the football pools problem, the alphabet A is ternary, since each match result being predicted has three possible outcomes: 'win' (for the home team, say), 'lose', or 'draw'.

For a Hamming code, the covering radius is 1. Indeed, for any perfect e-error-correcting code, the covering radius is e.

(4.2.1) PROPOSITION. *Let C be an e-error-correcting code. Then $\mathrm{cr}(C) \geq e$ with equality if and only if C is a perfect e-error-correcting code.*

PROOF. As C is an e-error-correcting code, the spheres of radius e around codewords are pairwise disjoint. Therefore the spheres of radius $e-1$ around codewords do not cover the whole space. Thus $\mathrm{cr}(C) > e-1$, whence $\mathrm{cr}(C) \geq e$. If we have equality, then we must have equality in the Sphere Packing Bound 2.2.6, hence the code is perfect. □

(4.2.2) PROPOSITION. *The covering radius of the linear code C is equal to the maximum weight of a coset leader.*

PROOF. The coset of the word $-\mathbf{x}$ consists of the sum of $-\mathbf{x}$ with each individual codeword of C, so the weights of the coset members give the distances of $\mathbf{x}$ from the various codewords. The minimal such weight is thus the distance of $\mathbf{x}$ from the code and also the weight of a coset leader. The maximum weight of a coset leader is therefore the largest distance of any word $\mathbf{x}$ from the code. □

As with $\mathrm{d}_{\min}$, the covering radius of a code is, in general, difficult to compute. The following problem, reminiscent of Problem 4.1.5, can be of great help.

(4.2.3) PROBLEM. *Let the $[n_1, k_1]$ linear code C_1 over F have generator matrix G_1, and let the $[n_2, k_2]$ linear code C_2 over F have generator matrix G_2. Consider the $[n_1+n_2, k_1+k_2]$ linear code C over F with generator matrix*

$$G = \begin{bmatrix} \mathbf{0} & G_1 \\ G_2 & * \end{bmatrix},$$

where the upper left $\mathbf{0}$ is a $k_1 \times n_2$ matrix of 0's and the lower right $*$ is an arbitrary $k_2 \times n_1$ matrix with entries from F.

Prove that $\mathrm{cr}(C) \leq \mathrm{cr}(C_1) + \mathrm{cr}(C_2)$.

4.3 First order Reed-Muller codes

In 1954, I.S. Reed and D.E. Muller introduced independently a class of binary codes of length 2^m, for any integer m, associated with Boolean logic. The first of these codes, for each m, fits naturally into the context of Hamming codes.

first order Reed-Muller code

A code dual to a binary extended Hamming code is called a *first order Reed-Muller code*, denoted $\mathrm{RM}(1, m)$ where m is the redundancy of the associated Hamming code. Any code that is equivalent to a first order Reed-Muller code is also first order Reed-Muller. We shall concentrate on the specific code $\mathrm{RM}(1, m)$ with generator matrix XL_m.

shortened first order Reed-Muller code
simplex code

The associated dual Hamming code is sometimes called a *shortened first order Reed-Muller code* or a *simplex code*. The dual Hamming code can be

easily recovered from $\mathrm{RM}(1,m)$. Indeed by first choosing all the codewords of $\mathrm{RM}(1,m)$ that have a 0 in their first coordinate position and then deleting this now useless coordinate, we find the dual Hamming code. This is clear when we consider how the matrix XL_m was constructed by bordering the matrix L_m, the generator matrix of the dual lexicographic Hamming code. (See page 51.)

Having earlier constructed the generator XL_m as a matrix in bordered block form, we now examine it again, but blocked in a different manner. Notice that $\mathrm{RM}(1,1) = \mathbb{F}_2^2$, $\mathrm{RM}(1,2)$ is the parity check code of length 4, and $\mathrm{RM}(1,3)$ is a self-dual extended $[8,4]$ Hamming code.

EXAMPLES.

$$\mathrm{XL}_1 = \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 1 \end{array}\right] \quad \text{and} \quad \mathrm{XL}_2 = \left[\begin{array}{cc|cc} 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array}\right]$$

$$\mathrm{XL}_3 = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

$$\mathrm{XL}_4 = \left[\begin{array}{cccccccc|cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

For i between 2^{m-1} and 2^m, the column m-tuple containing the binary representation of i is just that for $i - 2^{m-1}$ with its leading 0 replaced by a 1. Therefore, if we ignore the top row of XL_m, then the remaining m rows consist of an $m \times 2^{m-1}$ matrix repeated twice. Indeed this repeated matrix is nothing other than XL_{m-1}. We now observe the recursive construction:

$$\mathrm{XL}_m = \left[\begin{array}{c|c} 0 \cdots 0 & 1 \cdots 1 \\ \hline \mathrm{XL}_{m-1} & \mathrm{XL}_{m-1} \end{array}\right] .$$

(4.3.1) THEOREM. *For each m, the first order Reed-Muller code $\mathrm{RM}(1,m)$ is a binary linear $[2^m, m+1, 2^{m-1}]$ code.*

PROOF. Certainly $\mathrm{RM}(1,m)$ is linear of length 2^m, and its dimension $m+1$ is evident from the generator matrix XL_m. From their generator matrices XL_1 and XL_2, it is easy to see that $\mathrm{RM}(1,1)$ $(= \mathbb{F}_2^2)$ and $\mathrm{RM}(1,2)$ (the parity check code of length 4) both have minimum distance 2^{m-1}.

We now verify the minimum distance in general by induction, assuming that we already have $\mathrm{d_{min}}(\mathrm{RM}(1,m-1)) = 2^{m-2}$. Let C_1 be $\mathrm{RM}(1,m-1)$ with minimum distance $d_1 = 2^{m-2}$, and let C_2 be the repetition code of length 2^{m-1}, whose minimum distance is therefore $d_2 = 2^{m-1}$. The generator matrix XL_m for $\mathrm{RM}(1,m)$ is then constructed from the generators $G_1 = \mathrm{XL}_{m-1}$ and $G_2 = [1 \cdots 1]$ according to the recipe of Problem 4.1.5. Therefore, by that problem, we have

$$\mathrm{d_{min}}(\mathrm{RM}(1,m)) = \min(2d_1, d_2) = \min(2.2^{m-2}, 2^{m-1}) = 2^{m-1} ,$$

as claimed. □

(4.3.2) THEOREM. *The first order Reed-Muller code* $\mathrm{RM}(1,m)$ *consists of a unique word of weight* 0*, namely* $\mathbf{0}$*, a unique word of weight* 2^m*, namely* $\mathbf{1}$*, and* $2^{m+1}-2$ *words of weight* 2^{m-1}*.*

PROOF. The last row of the generator matrix XL_m is $\mathbf{1}$; so $\mathbf{0}$ and $\mathbf{1}$ are the unique codewords of weight 0 and 2^m, respectively. By Theorem 4.3.1 the linear code $\mathrm{RM}(1,m)$ has no codewords $\mathbf{c}$ of weight between 0 and 2^{m-1}, and so it also has no codewords $\mathbf{1}+\mathbf{c}$ of weight between 0 and 2^{m-1}. That is, it has no codewords of weight between 2^{m-1} and 2^m. Therefore all codewords other than $\mathbf{0}$ and $\mathbf{1}$ have weight exactly 2^{m-1}. □

(4.3.3) COROLLARY. *The dual of the binary Hamming code of redundancy* m *consists of a unique word of weight* 0*, namely* $\mathbf{0}$*, and* 2^m-1 *words of weight* 2^{m-1}*.*

PROOF. In recovering the dual Hamming code from $\mathrm{RM}(1,m)$, we shorten the code by taking all codewords that begin with 0 and then delete that position. In particular the codeword $\mathbf{1}$ of $\mathrm{RM}(1,m)$ does not survive. But by Theorem 4.3.2 all other nonzero codewords of $\mathrm{RM}(1,m)$ have weight 2^{m-1}. As only zeros are deleted, all the nonzero codewords of the dual Hamming code also will have weight 2^{m-1}. □

equidistant codes

These dual Hamming codes are *equidistant codes* in that distinct codewords are at a fixed distance from each other, here 2^{m-1}. They satisfy the Plotkin bound 2.3.8 with equality. (The proof of the Plotkin bound as given in Problem 3.1.5 compares the minimum distance with the average distance between codewords. For an equidistant code these are the same.)

For a binary word $\mathbf{x}\in\mathbb{F}_2^n$, consider the corresponding word $\mathbf{x}^*\in\{+1,-1\}^n$ gotten by replacing each 0 by the real number $+1$ and each 1 by -1.

(4.3.4) LEMMA. *If* $\mathbf{x},\mathbf{y}\in\mathbb{F}_2^n$*, then as vectors of real numbers* $\mathbf{x}^*\cdot\mathbf{y}^* = n-2\mathrm{d_H}(\mathbf{x},\mathbf{y})$*. In particular if* $\mathbf{x},\mathbf{y}\in\mathbb{F}_2^{2h}$ *with* $\mathrm{d_H}(\mathbf{x},\mathbf{y})=h$*, then* $\mathbf{x}^*\cdot\mathbf{y}^*=0$*.*

PROOF. The dot product of two ±1 vectors is the number of places in which they are the same minus the number of places where they are different. Here that is $(n-\mathrm{d_H}(\mathbf{x},\mathbf{y}))-\mathrm{d_H}(\mathbf{x},\mathbf{y})$. □

Let $\mathrm{RM}(1,m)^{\pm}$ be the code got by replacing each codeword $\mathbf{c}$ of $\mathrm{RM}(1,m)$ with its ±1 version $\mathbf{c}^*$. List the codewords of $\mathrm{RM}(1,m)^{\pm}$ as $\mathbf{c}_1^*, \mathbf{c}_2^*, \ldots, \mathbf{c}_{2^{m+1}}^*$.

(4.3.5) LEMMA. *If* $\mathbf{c}^*\in\mathrm{RM}(1,m)^{\pm}$ *then also* $-\mathbf{c}^*\in\mathrm{RM}(1,m)^{\pm}$*. We have*

$$\begin{array}{rcrl} \mathbf{c}_i^*\cdot\mathbf{c}_j^* & = & 2^m & \text{if } \mathbf{c}_i^*=\mathbf{c}_j^* \\ & = & -2^m & \text{if } \mathbf{c}_i^*=-\mathbf{c}_j^* \\ & = & 0 & \text{if } \mathbf{c}_i^*\neq\pm\mathbf{c}_j^* \ . \end{array}$$

PROOF. As $\mathbf{1} \in \mathrm{RM}(1,m)$ we have $(\mathbf{1}+\mathbf{c})^* = -\mathbf{c}^* \in \mathrm{RM}(1,m)^{\pm}$. By Theorem 4.3.2, if distinct $\mathbf{b},\mathbf{c} \in \mathrm{RM}(1,m)$ with $\mathbf{b} \neq \mathbf{1}+\mathbf{c}$, then $\mathrm{d_H}(\mathbf{b},\mathbf{c}) = 2^{m-1}$. The lemma follows from Lemma 4.3.4. □

We use this lemma as the basis of a decoding algorithm. When a vector $\mathbf{r}$ is received, calculate each of the dot products $\mathbf{r}\cdot\mathbf{c}_i^*$, for $i = 1,\ldots,2^{m+1}$. Then decode to that codeword $\mathbf{c}_j^*$ that maximizes the dot product.

In fact this can be done a little more efficiently. Arrange our listing of $\mathrm{RM}(1,r)$ so that $\mathbf{c}_{i+2^m}^* = -\mathbf{c}_i^*$, for each $i = 1,\ldots,2^m$. That is, the second half of the list is just the negative of the first half. In decoding, we calculate only those dot products from the first half $\mathbf{r}\cdot\mathbf{c}_i^*$, for $i = 1,\ldots,2^m$, and select that j that maximizes the absolute value $|\mathbf{r}\cdot\mathbf{c}_j^*|$. The received word $\mathbf{r}$ is then decoded to $\mathbf{c}_j^*$ if $\mathbf{r}\cdot\mathbf{c}_j^*$ is positive and to $-\mathbf{c}_j^*$ if $\mathbf{r}\cdot\mathbf{c}_j^*$ is negative.

We organize this as *Hadamard transform decoding*. Set $n = 2^m$, and let H_n be the $n \times n$ matrix whose i^{th} row is the codeword $\mathbf{c}_i^*$. The dot products of Lemma 4.3.5 then give

Hadamard transform decoding

$$H_n H_n^{\top} = n\, I_{n\times n}\,,$$

since the negative of a row of H_n is never a row. Upon receiving the vector $\mathbf{r}$, we calculate its Hadamard transform $\widehat{\mathbf{r}} = H_n\mathbf{r}^{\top}$. If $\widehat{r}_j$ is that coordinate of $\widehat{\mathbf{r}}$ that has the largest absolute value, then we decode $\mathbf{r}$ to $\mathbf{c}_j^*$ in case $\widehat{r}_j > 0$ or to to $-\mathbf{c}_j^*$ in case $\widehat{r}_j < 0$.

An important aspect of Hadamard transform decoding is that it is a soft decision algorithm rather than a hard decision algorithm. We need not require that the received vector $\mathbf{r}$ have entries ± 1. Its entries can be arbitrary real numbers, and the algorithm still works without modification.

EXAMPLE. Consider the code $\mathrm{RM}(1,3)^{\pm}$ which comes from $\mathrm{RM}(1,3)$ of length $n = 2^3 = 8$ with generator matrix XL_3. Let

$$H_8 = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix},$$

so $H_8 H_8^{\top} = 8\, I_{8\times 8}$. The codewords of $\mathrm{RM}(1,3)^{\pm}$ are then the rows of H_8 and their negatives.

Suppose we receive the vector $\mathbf{r} = (1,1,-1,1,-1,-1,1,1)$. This has Hadamard transform $H_8\mathbf{r}^{\top} = (2,-2,2,-2,2,-2,6,2)$. The entry with largest absolute value is $\widehat{r}_7 = 6 > 0$, so we decode to

$$\mathbf{c}_7^* = (+1,+1,-1,-1,-1,-1,+1,+1)\,.$$

If next we receive $\mathbf{r} = (-.7,1,0,-.8,-.9,1,.9,-1)$, then

$$H_8\mathbf{r}^{\top} = (-.5,-.9,1.3,-6.3,-.5,-.9,.9,1.3)\,.$$

The entry with largest absolute value is $\widehat{r}_4 = -6.3 < 0$, so we decode to

$$-\mathbf{c}_4^* = (-1, +1, +1, -1, -1, +1, +1, -1)\,.$$

(4.3.6) PROBLEM. *Assume that you are using the code* $\mathrm{RM}(1,3)^{\pm}$ *of the example. Use Hadamard transform decoding to decode the received word*

$$(.5\,,\ .4\,,\ -.6\,,\ .5\,,\ .6\,,\ -.3\,,\ .5\,,\ -.6\,)\ .$$

For any positive integer n, a ± 1 square matrix H of side n that satisfies

$$HH^{\top} = n\,I_{n\times n}$$

Hadamard matrix

is called a *Hadamard matrix*. If we take as a code the rows of H and their negatives, then Hadamard transform decoding will work exactly as described above. Such a code (or its $\{0,1\}$ counterpart) is called a *Hadamard code*.

Hadamard code

(4.3.7) PROBLEM. *Prove that a Hadamard matrix of side* n *must have* $n = 1, 2$ *or* n *a multiple of* 4. *(*REMARK. *It is a long-standing conjecture of combinatorial design theory that the converse of this problem is true: for each such* n*, there exists a Hadamard matrix.)*

Begin with a Hadamard code of side n. Choose those n codewords that start with $+1$, drop that position, and translate back to a $\{0,1\}$ code. The result is a binary code of length $n-1$ and size n which is equidistant of distance $n/2$. A code constructed in this fashion is a *shortened Hadamard code*. Starting with the matrix H_8 of the example above, we recover from $\mathrm{RM}(1,3)$ and $\mathrm{RM}(1,3)^{\pm}$ the $[7,3]$ dual Hamming code.

shortened Hadamard code

(4.3.8) PROBLEM. *Let* h *be a positive integer. Let* C *be a binary equidistant code of length* $2h-1$ *and size* $2h$ *with distance* h.

(*a*) *Prove that* C *is a shortened Hadamard code.*

(*b*) *Prove that* C *meets the Plotkin bound 2.3.8 with equality.*

Although any Hadamard matrix can be used to design a code that allows Hadamard transform decoding, there are certain advantages to be gained from using those matrices that come from Reed-Muller codes as described. The existence of a soft decision algorithm is good, but we hope to implement it as efficiently as possible. Consider decoding using the matrix H_8 of the example. Each decoding process requires 63 operations, 56 additions for calculating the 8 dot products and 7 comparisons among the answers. (By convention the operation of negation is considered to make no contribution.) Certain annoying repetitions are involved. For instance, both the second and the sixth rows of H_8 begin $+1, -1, +1, -1$; so the corresponding calculation $r_1 - r_2 + r_3 - r_4$ is made twice during the decoding process. Can this and other patterns within H_8 be exploited? The answer is "yes," and it is this fact that makes a matrix derived from $\mathrm{RM}(1,m)$ a better choice than other Hadamard matrices with the same dimensions.

Let $H_1 = [1]$, a 1×1 Hadamard matrix, and define recursively a $2^{m+1} \times 2^{m+1}$ matrix in block form

$$H_{2^{m+1}} = \begin{bmatrix} +H_{2^m} & +H_{2^m} \\ +H_{2^m} & -H_{2^m} \end{bmatrix}.$$

Then

$$H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix},$$

and

$$H_4 = \left[\begin{array}{cc|cc} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ \hline +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{array}\right] = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}.$$

The matrix H_8 is that of the example. This construction can be continued, for all m. The matrix H_{2^m} produced is a Hadamard matrix associated with $\mathrm{RM}(1,m)^{\pm}$ and the Reed-Muller code $\mathrm{RM}(1,m)$ whose generator matrix is XL_m. The recursive construction of H_{2^m} is related to that of XL_m and admits a streamlined implementation of decoding for $\mathrm{RM}(1,m)$ and $\mathrm{RM}(1,m)^{\pm}$, using the so-called Fast Hadamard Transform or FHT algorithm. For instance, FHT decoding of $\mathrm{RM}(1,3)^{\pm}$ can be achieved with 31 operations rather than the 63 counted previously.

The Reed-Muller codes in general, and the code $\mathrm{RM}(1,3)$ in particular, are important codes that often arise as constituents of larger codes. It is therefore worthwhile to have decoding algorithms that are as efficient as possible. Sun and Van Tilborg have given a soft decision algorithm for $\mathrm{RM}(1,3)$ that is related to FHT decoding but only needs, on the average, 14 operations with worst-case performance of 17 operations.