On the Weights of Linear Codes and their Dual

Lauren Bolcar

Fall 2019

MA 398, Dr. Molitierno

Faculty Mentor: Dr. Boyle

Abstract

Codes were invented to detect and correct transmission errors caused by noise on a communication channel. In this paper, we will look at linear codes as well as the dual code and the matrices that allow us to convert between the two. For linear codes, the error correcting capabilities of a code are determined by the weights, in particularly the minimum weight, of the codewords. We will explore these weights and how to find their minimum value as well as introduce the MacWilliams Theorem, which connects the weights of a code to the weights of its dual.

1 Introduction to Coding Theory

When information or messages are transmitted to some receiver, it is done so through a communication channel. Ideally, at the end of this transmission, the information obtained at the receiving end is the same information that was sent into the channel. In reality, however, this is not always the case, since it is likely that errors will occur during transmission. This is because our messages are sent over a noisy channel, which simply means that some sort of distraction of interruption happened in the channel that caused what was received to be different than what was sent. As a result, codes were invented to detect and correct transmission errors caused by noise on the channel.

2 Background

DEFINITION 2.1 A *code* C over an *alphabet* A is a subset of $A^n := A \times ... \times A$ (n copies).

For the purpose of this paper, A will always be a finite field. In particular, we will consider the binary field $\mathbb{F}_2 := \{0, 1\}$, where addition and multiplication are done modulo 2.

DEFINITION 2.2 Elements of a code are called *codewords*, and the *length* of the code is n. When A is a field, $C \subset A^n$ is called a *linear code* if it is a vector

subspace of A^n. The *dimension* k of C is defined to be the dimension of C as a vector space over A. Together, n and k are called the *parameters* of C. We call the code an $[n, k]$ code.

Recall that a vector subspace is closed under linear operations. Thus, if $\mathbf{x}, \mathbf{y} \in C$, then $a\mathbf{x} + b\mathbf{y} \in C$ for all $a, b \in A$. Also, recall that the dimension of a vector space is, by definition, the number of vectors in a basis, or, equivalently, the number of nonzero rows when in reduced row echelon form.

For a linear code C of length n and dimension k over $\mathbb{F}_2$, there are k basis elements for C, each of which is a vector of length n.

DEFINITION 2.3 Let C be a linear $[n, k]$ code. A *generator matrix* for C is a $k \times n$ matrix such that the rows are the basis vectors for C.

For example, consider the 3×6 generator matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{2.1}$$

Since G is in reduced row echelon form, its rows are the basis vectors of some linear code C. We see that there are 3 vectors in the basis, each of length 6, meaning that we have a $[6, 3]$ code with length $n = 6$ and dimension $k = 3$.

The code C is the set of all possible linear combinations of the rows of G. If

G is a generator matrix for C, then C is exactly the set

$$\{\mathbf{u}G \mid \mathbf{u} \in \mathbb{F}_2^k\}, \tag{2.2}$$

where $\mathbf{u}$ is all the possible $1 \times k$ binary row message vectors and G is of the form

$$G = [I_k|P]. \tag{2.3}$$

In (2.3), I_k is the $k \times k$ identity matrix and P is a $k \times (n-k)$ matrix of 0's and 1's.

THEOREM 2.4 *Let a linear code C be a vector space over $\mathbb{F}_2$. If dim(C) = k, then C has 2^k codewords.*

Proof: Suppose $\dim(C) = k$ and let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ be a basis for C. Then, $C = \lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2 + \ldots + \lambda_k\mathbf{x}_k$ where $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{F}_2$. Since $|\mathbb{F}_2| = 2$, there are exactly 2 choices for each $\lambda_1, \lambda_2, \ldots \lambda_k$. Each choice gives a different word and so C has exactly 2^k codewords. □

Continuing with our example, then, using the matrix in (2.1), C has $2^3 = 8$ codewords and is the set as described in (2.2). Each codeword can therefore be found by multiplying the generator matrix G on the left by a possible message vector. For instance, using the message vector

$$\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix},$$

we get that one codeword is

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

The other codewords would similarly be obtained by doing the multiplication with the 7 other possible 1×3 message vectors. The code C then contains the following codewords:

$$100110 \quad 010011 \quad 001101 \quad 111000 \tag{2.4}$$

$$110101 \quad 101011 \quad 011110 \quad 000000.$$

Notice also that, in accordance with (2.3),

$$I_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathrm{P} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

DEFINITION 2.5 If C is an $[n, k]$ linear code over $\mathbb{F}_2$, its *dual code* $C^\perp$ is the set of vectors which are orthogonal to all codewords of C:

$$C^\perp = \{\mathbf{v} \mid \mathbf{u} \cdot \mathbf{v} = 0 \text{ for all } \mathbf{u} \in C\}.$$

The dual code $C^\perp$ is an $[n, n-k]$ code, which can be proven using the following lemmas and the Rank-Nullity Theorem from Linear Algebra [5].

LEMMA 2.6 *$C^\perp$ is a linear code.*

Proof: If $\mathbf{y},\ \mathbf{y}' \in C^\perp$, then $\mathbf{x} \cdot \mathbf{y} = \mathbf{x} \cdot \mathbf{y}' = 0$ for all $\mathbf{x} \in C$. This implies that $\mathbf{x} \cdot (\mathbf{y} + \mathbf{y}') = \mathbf{x} \cdot \mathbf{y} + \mathbf{x} \cdot \mathbf{y}' = 0$ for all $\mathbf{x} \in C$ and that $\mathbf{x} \cdot (\lambda \mathbf{y}) = \lambda(\mathbf{x} \cdot \mathbf{y}) = \lambda(0) = 0$ for all $\mathbf{x} \in C$. Thus, $C^\perp$ is a linear code. □

LEMMA 2.7 *Let G be a generator matrix of the code C. The dual code $C^\perp =$ Null G, or, equivalently, $\mathbf{w} \in C^\perp$ if and only if $G\mathbf{w}^T = 0$.*

Proof: Let g_i denote the rows of G for $1 \leq i \leq k$. Then, $g_1,\ \ldots\ , g_k$ are the codewords that form a basis for C. We will show $G\mathbf{w}^T = 0$ if and only if $g_i \cdot \mathbf{w}$ for all i.

($\Leftarrow$) Let $\mathbf{w} \in C^\perp$. Then, $\mathbf{u} \cdot \mathbf{w} = 0$ for all $\mathbf{u} \in C$. Since the rows of G are the codewords of C, $g_i \cdot \mathbf{w} = 0$ for $i = 1,\ \ldots\ k$. So, $G\mathbf{w}^T = 0$.

($\Rightarrow$) Assume that $g_1,\ \ldots\ g_k$ are the rows of G and that $G\mathbf{w}^T = 0$. Then, $g_i \cdot \mathbf{w} = 0$ for all i. Since the rows of G form a basis of C, if $\mathbf{x}$ is any codeword of C, then $\mathbf{x} = \sum_{i=1}^{k} a_i g_i$ for some scalers $a_i \in \mathbb{F}_2$. So,

$$\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{k} a_i(\mathbf{w} \cdot g_i) = \sum_{i=1}^{k} a_i \cdot 0 = 0.$$

So, $\mathbf{w}$ is orthogonal to every codeword in C, which implies that $\mathbf{w} \in C^\perp$.

Thus, $C^\perp =$ Null G. □

THEOREM 2.8 *If C is an $[n, k]$ linear code over F_2, then $C^{\perp}$ is an $[n, n-k]$ code over $\mathbb{F}_2$.*

Proof: Suppose C is an $[n, k]$ linear code over $\mathbb{F}_2$. Then, C is a subspace of $\mathbb{F}_2^n$ and the dimension of C is k. Let G be a generator matrix for C. By the Rank-Nullity Theorem from Linear Algebra, Rank G + dim(Null G) = n. Since, by definition, Rank G is equal to the number of nonzero rows when G is in row reduced form, which is equal to dim C, then Rank $G = k$. So, k + dim(Null G) = n implies that dim(Null G) = $n - k$.

By Lemma 2.6, $C^{\perp}$ is a linear code, so $C^{\perp}$ is a subspace of $\mathbb{F}_2^n$. Since $C^{\perp}$ is the null space of G by Lemma 2.7 and dim(Null G) = $n - k$, then dim $C^{\perp}$ = dim(Null G) = $n - k$. So, $C^{\perp}$ is an $n - k$ dimensional subspace of $\mathbb{F}_2^n$. Thus, $C^{\perp}$ is an $[n, n-k]$ code. □

DEFINITION 2.9 A *parity check matrix H* for an $[n, k]$ code is an $(n-k) \times n$ matrix which is a generator matrix of $C^{\perp}$, given by

$$H = [P^T \mid I_{n-k}]. \tag{2.6}$$

THEOREM 2.10 *If G is the generator matrix of C given by $[I_k|P]$, then H is the generator matrix of $C^{\perp}$, or, equivalently, the parity check matrix of C.*

Proof: Let G be the generator matrix of C given by $[I_k|P]$, where, as a reminder, I_k is the $k \times k$ identity matrix and P is a $k \times (n-k)$ matrix. Assume that $H = [P^T \mid I_{n-k}]$ and rank $H = n - k$. Then,

$$GH^T = [I_k \ P] \begin{bmatrix} P \\ I_{n-k} \end{bmatrix} = I_k P + P I_{n-k}.$$

When in the binary case, though, $P = -P$ as $-1 = 1$ and $0 = 0$ in modulo 2. So,

$$GH^T = I_k(-P) + P I_{n-k} = -P + P = 0.$$

Thus, $GH^T = 0$ implies that the rows of H are orthogonal to the rows of G. Since the rows of G represent the codewords of C, this indicates that the rows of H are in $C^\perp$. We also know that the dim $C^\perp = n - k$ by Theorem 2.8 and that the dimension of a code equals the rank of its generator matrix. Thus, since rank $H = n - k$ and dim $C^\perp = n - k$, H is the generator matrix for $C^\perp$. □

If we have the generator matrix of C, (2.6) allows us to quickly find the parity check matrix of C. For example, using (2.3) and (2.5), we obtain

$$P^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \mathrm{I}_{6-3} = \mathrm{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.7}$$

which by (2.6) indicates that

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{2.8}$$

Since the parity check matrix H is the generator matrix of $C^{\perp}$, the dual code is the set of all linear combinations of the rows of H. We can then find the codewords of $C^{\perp}$ by similarly multiplying all the possible $1 \times (n-k)$ binary row vectors and H. Thus, finishing our example, the dual code $C^{\perp}$ then contains the following codewords:

$$101100 \quad 110010 \quad 011001 \quad 000111 \tag{2.9}$$

$$011110 \quad 110110 \quad 101011 \quad 000000.$$

Additionally, the code C can also be described by its parity check matrix H. In this scenario,

$$C = \{\mathbf{x} \in \mathbb{F}_2^n \mid H\mathbf{x}^T = 0\}. \tag{2.10}$$

As such, the parity check matrix H allows us to check whether a received word is in the code C. If the vector is a valid codeword, we assume that the message was transmitted correctly. In other words, if $H\mathbf{x}^T = 0$, then $\mathbf{x}$ is a codeword of C and we conclude it is the same codeword originally sent through the channel. On the other hand, if $H\mathbf{x}^T \neq 0$, then $\mathbf{x}$ is not a codeword of C and so we know that an error occurred in transmission.

3 Weights of Linear Codes and Some Results

DEFINITION 3.1 Let $\mathbf{x}$ and $\mathbf{y}$ be codewords of length n over $\mathbb{F}_2$. The (*Hamming*)*distance* from $\mathbf{x}$ to $\mathbf{y}$, denoted by $d(\mathbf{x}, \mathbf{y})$, is defined to be the number of coordinate places in which $\mathbf{x}$ and $\mathbf{y}$ differ. If $\mathbf{x} = x_1 \dots x_n$ and $\mathbf{y} = y_i \dots y_n$, then

$$d(\mathbf{x}, \mathbf{y}) = d(x_1, y_1) + \ldots + d(x_n, y_n),$$

where x_i and y_i are regarded as words of length 1, and

$$d(x_i, y_i) = \begin{cases} 1 \text{ if } x_i \neq y_i \\ 0 \text{ if } x_i = y_i. \end{cases}$$

For example, using two codewords from the code C in (2.4),

$$d(100110, 010011) = 1 + 1 + 0 + 1 + 0 + 1 = 4.$$

DEFINITION 3.2 The *weight* of a vector $\mathbf{x} \in \mathbb{F}_2^n$, denoted by $wt(\mathbf{x})$, is the number of nonzero coordinates in x; i.e.,

$$wt(\mathbf{x}) = d(\mathbf{x}, \mathbf{0}),$$

where $\mathbf{0}$ is the zero word.

LEMMA 3.3 *If* $\mathbf{x}$, $\mathbf{y} \in \mathbb{F}_2^n$, *then* $d(\mathbf{x},\mathbf{y}) = wt(\mathbf{x} - \mathbf{y})$.

Proof: Let $\mathbf{x}$, $\mathbf{y} \in \mathbb{F}_2^n$. By definition, $d(\mathbf{x}, \mathbf{y})$ is the number of places where $\mathbf{x}$ and $\mathbf{y}$ differ. Then, the vector $\mathbf{x} - \mathbf{y}$ will have a 1 precisely in the places where

$\mathbf{x}$ and $\mathbf{y}$ differ and a 0 in the places where they are the same. In other words, $d(\mathbf{x}, \mathbf{y})$ is equal to the number of places where there is a 1 in the vector $\mathbf{x} - \mathbf{y}$. But, the number of places with a 1 is, by definition, the weight of that vector. So, $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y})$. □

DEFINITION 3.4 The *minimum distance* of a code C, denoted by d or $d(C)$, is

$$d = d(C) = \min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}.$$

DEFINITION 3.5 The *minimum weight* of a code C, denoted $wt(C)$, is

$$wt(C) = \min\{wt(\mathbf{x}) \mid \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}$$

THEOREM 3.6 *If C is a linear code in $\mathbb{F}_2^n$, then the minimum distance is the minimum weight of any nonzero codeword. In other words, $d(C) = wt(C)$.*

Proof: Let $\mathbf{x}$ and $\mathbf{y}$ be two codewords in C such that $d(\mathbf{x}, \mathbf{y}) = d(C)$. Then, by Lemma 3.3, $d(C) = d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y}) \geq wt(C)$, since $\mathbf{x} - \mathbf{y} \in C$. Conversely, there is a codeword $\mathbf{z} \in C \setminus \{\mathbf{0}\}$ such that $wt(C) = wt(\mathbf{z})$. So, $wt(C) = wt(\mathbf{z}) = d(\mathbf{z}, \mathbf{0}) \geq d(C)$, since $\mathbf{0} \in C$. Thus, $d(C) = wt(C)$. □

As a result of Theorem 3.6, the minimum distance and minimum weight of a linear code C can be used interchangeably.

In addition to the length and dimension of a code C, the minimum distance d, or, equivalently, the minimum weight d, is another important parameter of a linear code as it determines the code's error-correcting capabilities. A linear code with minimum weight d can correct $\lfloor \frac{d-1}{2} \rfloor$ errors, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. If d is even, the code can simultaneously correct $\lfloor \frac{d-2}{2} \rfloor$ errors and detect $\frac{d}{2}$ errors. From the formulas, it is clear that to correct a single error d needs to be at least 3. The number of errors that a code can correct is a measure of how good a code is. As such, determining the minimum weight of a code, as well as learning about the weights in general, is crucial to knowing more about a code.

One way to find the minimum weight d of a linear code C is to examine the weights of all the nonzero codewords. Looking at the codewords of the code C as in (2.4), for example, we determine the weights of each codeword:

$$wt(100110) = 3 \quad wt(010011) = 3 \quad wt(001101) = 3 \quad wt(111000) = 3 \qquad (3.1)$$

$$wt(110101) = 4 \quad wt(101011) = 4 \quad wt(011110) = 4 \quad wt(000000) = 0.$$

By examining each of the possible nonzero weights, it is clear that $d = wt(C) = 3$.

For the sake of completeness, we can similarly determine the minimum weight of the dual code $C^{\perp}$, now denoted by $d^{\perp}$. The weights of the dual code codewords

from (2.9) are as follows:

$$wt(101100) = 3 \quad wt(110010) = 3 \quad wt(011001) = 3 \quad wt(000111) = 3 \qquad (3.2)$$

$$wt(011110) = 4 \quad wt(110110) = 4 \quad wt(101011) = 4 \quad wt(000000) = 0.$$

Again, the minimum weight of the nonzero codewords in $C^{\perp}$ is 3.

Notice, interestingly enough, that the weights of all the codewords in both (3.1) and (3.2) are split exactly in half in terms of even verses odd weights. However, consider, briefly, the $[3, 2]$ code C with generator matrix and parity check matrix given by

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \mathrm{H} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \qquad (3.3)$$

The codewords of C with their respective weights are then

$$wt(101) = 2 \quad wt(011) = 2 \quad wt(110) = 2 \quad wt(000) = 0 \qquad (3.4)$$

In (3.4), the minimum weight of the nonzero codewords is 2.

REMARK 3.7 The dual code of this $[3, 2]$ code has only the codewords 000 and 111, which have weights 0 and 3, respectively.

Notice, though, that in (3.4) all of the four codewords have even weight. We have the following result.

THEOREM 3.8 *In a binary linear code, either all the codewords have even weight, or exactly half have even weight and half have odd weight.*

To prove this theorem, we first need the following three lemmas.

LEMMA 3.9 *If* $\mathbf{x}$, $\mathbf{y} \in \mathbb{F}_2^n$, *then* $wt(\mathbf{x}+\mathbf{y}) = wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x} \cap \mathbf{y})$ *where* $\mathbf{x} \cap \mathbf{y}$ *is the vector* $(x_1y_1, x_2y_2, \ldots, x_ny_n)$, *which has* 1*'s only in the positions where both* $\mathbf{x}$ *and* $\mathbf{y}$ *have* 1*'s.*

Proof: Note that $wt(\mathbf{x}) - wt(\mathbf{x} \cap \mathbf{y})$ is the number of positions i such that $x_i = 1$ but $y_i = 0$. Similary, $wt(\mathbf{y}) - wt(\mathbf{x} \cap \mathbf{y})$ is the number of positions i such that $y_i = 1$ but $x_i = 0$. Then,

$$wt(\mathbf{x}) - wt(\mathbf{x} \cap \mathbf{y}) + wt(\mathbf{y}) - wt(\mathbf{x} \cap \mathbf{y}) = wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x} \cap \mathbf{y})$$

is the number of places where $\mathbf{x}$ and $\mathbf{y}$ differ. By definition, the number of places where $\mathbf{x}$ and $\mathbf{y}$ differ is the distance between $]bfx$ and $\mathbf{y}$. So, $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x} \cap \mathbf{y})$. But, by Lemma 3.3, $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y}) = wt(\mathbf{x} + \mathbf{y})$ since $-y = y$ in $\mathbb{F}_2$. So, $wt(\mathbf{x} + \mathbf{y}) = wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x}\ cap\mathbf{y})$. □

LEMMA 3.10 *Let* $\mathbf{x}$, $\mathbf{y} \in \mathbb{F}_2^n$. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *both have odd weights, then* $\mathbf{x} + \mathbf{y}$ *has even weight.*

Proof: Let $\mathbf{x}$ and $\mathbf{y}$ have odd weight. Then $wt(\mathbf{x}) = 2i + 1$ and $wt(\mathbf{y}) = 2j + 1$ for some i, $j \in \mathbb{Z}$. Let k be the number of positions where both $\mathbf{x}$ and $\mathbf{y}$ have 1's,

or $wt(\mathbf{x} \cap \mathbf{y})$ as in Lemma 3.9. Then,

$$\begin{aligned} wt(\mathbf{x}+\mathbf{y}) &= wt(\mathbf{x}) + wt(\mathbf{y}) - 2k \\ &= (2i+1) + (2j+1) - 2k \\ &= 2i + 2j - 2k + 2 \\ &= 2(i+j-k+1). \end{aligned}$$

So, $wt(\mathbf{x}+\mathbf{y})$ is even. □

LEMMA 3.11 *Let* $\mathbf{x}$, $\mathbf{y} \in \mathbb{F}_2^n$. *If one has odd weight and the other has even weight, then* $\mathbf{x}+\mathbf{y}$ *has odd weight.*

Proof: Without loss of generality, let $\mathbf{x}$ have odd weight and $\mathbf{y}$ have even weight. Then $wt(\mathbf{x}) = 2i+1$ and $wt(\mathbf{y}) = 2j$ for some $i, j \in \mathbb{Z}$. Let k be the number of positions where both $\mathbf{x}$ and $\mathbf{y}$ have 1's, or $wt(\mathbf{x} \cap \mathbf{y})$ as in Lemma 3.9. Then,

$$\begin{aligned} wt(\mathbf{x}+\mathbf{y}) &= wt(\mathbf{x}) + wt(\mathbf{y}) - 2k \\ &= (2i+1) + (2j) - 2k \\ &= 2i + 2j - 2k + 1 \\ &= 2(i+j-k) + 1. \end{aligned}$$

Thus, $wt(\mathbf{x}+\mathbf{y})$ has odd weight. □

Using the preceding lemmas, we can now prove Theorem 3.8.

Proof: (of Theorem 3.8) We will consider the following cases.

Case 1: Assume all the codewords have even weight. Then, the proof is complete.

Case 2: Assume that not all of the codewords have even weight. Then at least one codeword has odd weight. Let $\mathbf{x} \in C$ be a codeword with odd weight.

Let n be the number of even weight codewords in C and let m be the number of odd weight codewords in C. We want to show that $n = m$.

Recall that since C is linear, adding $\mathbf{x}$ to any other codeword results in another codeword of C. By Lemma 3.11, then, adding $\mathbf{x}$ to each of the n codewords with even weights gives n codewords with odd weights. So, the number of odd weight codewords is at least n, ie. $n \leq m$. By Lemma 3.10, adding $\mathbf{x}$ to each of the m codewords with odd weights gives m codewords with even weights. So, there are at least as many codewords with even weight as there are codewords with odd weights, ie. $n \geq m$.

Since $n \leq m$ and $n \geq m$, it must be the case that $n = m$. Thus, between the two cases, either all the codewords have even weight, or half the codewords have even weight and half the codewords have odd weight. □

Thus, in terms of the minimum weight, a code with all words of even weight will always have an even minimum weight. On the other hand, if half are even weights and half are odd weights, then the minimum weight can be either even or odd, depending on which value is smaller.

Alternatively, we can also determine the minimum distance, or weight, d of a code C using the parity check matrix H.

THEOREM 3.12 *Let H be the parity check matrix of a linear code C. A linear code has minimum weight d if and only if any $d-1$ columns of H are linearly independent and there exists some d columns that are linearly dependent.*

Proof: Let C be a linear code. Then, $wt(C) = d(C) = d$. So, there must exist some codeword in C with weight d. Suppose that the vector $\mathbf{u}$ is a codeword in C such that $wt(\mathbf{u}) = d$. Since $\mathbf{u} \in C$ implies that $H\mathbf{u}^T = 0$ and $\mathbf{u}$ has d nonzero components, then there are some d columns of H that are linearly dependent.

For the other side, suppose that there are $d-1$ linearly dependent columns in H. Then, there must exist a nonzero vector $\mathbf{v} \in C$ such that $wt(\mathbf{v}) = d-1$. This, however, contradicts the fact that the minimum weight of C is d. So, any $d-1$ columns of H are linearly independent. □

As a consequence of Theorem 3.12, we have the following special cases. For one, the minimum weight of a code C is 1 if H has a column of all zeros. But, now suppose that H does not have a zero column. Then, the minimum weight of C is at least 2. By our theorem, the minimum weight is equal to 2 if H has two columns that are linearly dependent. When working in binary, H has two linearly dependent columns only when two columns are identical. For example, consider

again the $[3, 2]$ code C in (3.3) with parity check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

We see that H has two identical columns and so the two columns are linearly dependent since $1 + 1 = 0$ modulo 2. So, the minimum weight d is 2, which is what we found earlier by looking at all the weights of the nonzero codewords of C.

As a result, if H has no columns of all zeros and all columns are distinct, the the minimum weight of a binary code C is at least 3. This was the case for the $[6, 3]$ code C in (2.4) with parity check matrix

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

We see by inspection that there is no zero column and no two identical columns, so any two columns of H are linearly independent. However, columns 1, 2 and 3 sum to $\mathbf{0}$, and thus are linearly dependent. Since any two columns of H are linearly independent and some three columns of H are linearly dependent, $d = 3$, which is again what was found using our first method for finding minimum weights.

In the above cases, only when d was equal to 3 was the code able to correct an error. It becomes clear then that a code C can correct more errors as d increases.

So, to be able to maximize the amount of errors C can correct, d needs to be as large as possible with respect to n and k.

THEOREM 3.13 *If C is an $[n, k]$ code with $wt(C) = d$, then $d \leq n - k + 1$.*

Proof: Let C be an $[n, k]$ code with minimum weight d. Let H be a parity check matrix of C. By definition, H is an $(n - k) \times n$ matrix of rank $n - k$. So, at most $n - k + 1$ columns of H are linearly dependent. By Theorem 3.12, since the minimum weight is d, there are d columns of H that are linearly dependent. Thus, $d \leq n - k + 1$. □

4 MacWilliams Theorem

In Section 3, our main focus was on determining the minimum weight of a code. However, looking solely at the minimum weight reveals nothing about the weights of the other codewords in the code. When we want to know about the other codewords of a code, we instead look at the weight distribution of the entire code.

DEFINITION 4.1 The *weight distribution* of a code of length n lists the number of codewords of each possible weight i for $0 \leq i \leq n$.

The most convenient way to express the weight distribution of a code is through its weight enumerator.

DEFINITION 4.2 The *weight enumerator* of C is defined to be

$$W_C(x,y) = \sum_{u \in C} x^{n-wt(u)} y^{wt(u)} = \sum_{i=0}^{n} A_i x^{n-i} y^i, \tag{4.1}$$

where A_i denotes the number of codewords of weight i in C.

DEFINITION 4.3 The *weight enumerator* of $C^\perp$ is defined to be

$$W_{C^\perp}(x,y) = \sum_{u \in C^\perp} x^{n-wt(u)} y^{wt(u)} = \sum_{i=0}^{n} A_i^\perp x^{n-i} y^i, \tag{4.2}$$

where $A_i^\perp$ denotes the number of codewords of weight i in $C^\perp$.

In both Definition 4.2 and 4.3, x and y are indeterminates while $W_C(x,y)$ and $W_{C^\perp}(x,y)$ are homogeneous polynomials of degree n in x and y. The coefficients of the terms of the polynomials determine the number of words of each weight from zero to n.

Using the weights found for the code C and its dual code in (3.4) and (3.7), respectively, we can find their weight enumerators. The code C has the weight distribution $A_1 = A_3 = 0$, $A_0 = 1$, and $A_2 = 3$ so the weight enumerator of C is

$$W_C(x,y) = 1x^3y^0 + 0x^2y^1 + 3x^1y^2 + 0x^0y^3 \tag{4.3}$$

$$= x^3 + 3xy^2. \tag{4.4}$$

The weight enumerator indicates that the code C has one word of weight 0 and three words of weight 2, which matches the weight distribution of the code.

Similarly, the dual code $C^{\perp}$ has the weight distribution $A_1^{\perp} = A_2^{\perp} = 0$, $A_0^{\perp} = 1$, and $A_3^{\perp} = 1$. The weight enumerator of $C^{\perp}$ is then

$$W_{C^{\perp}}(x, y) = 1x^3y^0 + 0x^2y^1 + 0x^1y^2 + 1x^0y^3 \tag{4.5}$$

$$= x^3 + y^3. \tag{4.6}$$

Again, as the weight distribution also indicated, the weight enumerator reveals that there is one codeword of weight 0 and one codeword of weight 3.

Florence Jessie MacWilliams, however, discovered that the distribution of a code C uniquely determines the distribution of its dual code. In fact, she found that $W_{C^{\perp}}(x, y)$ is given by a linear transformation of $W_C(x, y)$. Her result has since become known as the *MacWilliams Theorem for binary linear codes.*

THEOREM 4.4 *(MacWilliams Theorem) If C is an [n,k] binary linear code with dual code* $C^{\perp}$ *then*

$$W_{C^{\perp}}(x, y) = \frac{1}{|C|} W_C(x + y, x - y), \tag{4.7}$$

where $|C| = 2^k$ *is the number of codewords in C. Equivalently,*

$$\sum_{j=0}^{n} A_j^{\perp} x^{n-j} y^j = \frac{1}{|C|} \sum_{i=0}^{n} A_i (x + y)^{n-i} (x - y)^i, \tag{4.8}$$

or

$$\sum_{v \in C^{\perp}} x^{n-wt(v)} y^{wt(v)} = \frac{1}{|C|} \sum_{u \in C} (x + y)^{n-wt(u)} (x - y)^{wt(u)}. \tag{4.9}$$

The proof of Theorem 4.4 is attributed to MacWilliams [1] and relies on the following lemma.

LEMMA 4.5 *Let $f : \mathbb{F}_2^n \to A$ be any mapping from $\mathbb{F}_2^n$ into a vector space A over the complex numbers. Define the Hadamard transform $\hat{f}$ of f by*

$$\hat{f}(u) = \sum_{v \in \mathbb{F}_2^n} (-1)^{u \cdot v} f(v), u \in \mathbb{F}_2^n. \tag{4.10}$$

Then for a $[n, k]$ binary linear code C,

$$\sum_{v \in C^\perp} f(v) = \frac{1}{|C|} \sum_{u \in C} \hat{f}(u). \tag{4.11}$$

Proof: Let C be a binary linear code. Then,

$$\begin{aligned} \sum_{u \in C} \hat{f}(u) &= \sum_{u \in C} \sum_{v \in \mathbb{F}_2^n} (-1)^{u \cdot v} f(v) \\ &= \sum_{v \in \mathbb{F}_2^n} f(v) \sum_{u \in C} (-1)^{u \cdot v} \end{aligned}$$

after substituting in the definition of $\hat{f}(u)$ and reorganizing the terms.

Now, if $v \in C^\perp$. then $u \cdot v$ is always zero by definition of the dual code, and the inner sum, or $\sum_{u \in C} (-1)^{u \cdot v}$ is $|C|$. This is because $(-1)^0 = 1$ and so, when summed across every codeword of C, the result is the number of codewords in total, or $|C|$. But if $v \neq C^\perp$, then $u \cdot v = 0$ and 1 equally often and so $(-1)^{u \cdot v}$ equals 1 and -1 equally often, making the inner sum zero when all the results are added together. Therefore,

$$\sum_{u \in C} \hat{f}(u) = |C| \sum_{v \in C^\perp} f(v).$$

□

With the lemma, we can now prove the MacWilliams Theorem for binary linear codes.

Proof: (of Theorem 4.4) In Lemma 4.11, let A be the set of polynomials in x, y with complex coefficients, and let

$$f(v) = x^{n-wt(v)}y^{wt(v)}. \tag{4.12}$$

Then, substituting (4.12) into (4.10),

$$\hat{f}(u) = \sum_{v\in\mathbb{F}_2^n}(-1)^{u\cdot v}x^{n-wt(v)}y^{wt(v)}. \tag{4.13}$$

Now, let $u = (u_1 \dots u_n)$ and $v = (v_1 \dots v_n)$. Then (4.13) becomes

$$\begin{aligned}\hat{f}(u) &= \sum_{v\in\mathbb{F}_2^n}(-1)^{u_1v_1+\dots u_nv_n}\prod_{i=1}^{n}x^{1-v_i}y^{v_i}\\ &= \sum_{v_1=0}^{1}\sum_{v_2=0}^{1}\cdots\sum_{v_n=0}^{1}\prod_{i=1}^{n}(-1)^{u_iv_i}x^{1-v_i}y^{v_i}\\ &= \prod_{i=1}^{n}\sum_{v=0}^{1}(-1)^{u_iv}x^{1-v}y^{v}.\end{aligned}$$

If $u_i = 0$, the inner sum is $x + y$. If $u_i = 1$, the inner sum is $x - y$. Thus,

$$\hat{f}(u) = (x+y)^{n-wt(u)}(x-y)^{wt(u)}. \tag{4.14}$$

Therefore, by Lemma 4.11, using (4.12) and (4.14),

$$\sum_{v\in C^\perp}x^{n-wt(v)}y^{wt(v)} = \frac{1}{|C|}\sum_{u\in C}(x+y)^{n-wt(u)}(x-y)^{wt(u)},$$

which equals (4.9) and thus the theorem is proved. □

The MacWilliams Theorem gives us an alternative to finding the weight distribution and enumerator of the dual code using only information about C. So, for example, using the weight distribution of the $[3, 2]$ code found in (4.3), and assuming that all information about the dual code is unknown, the MacWilliams Theorem allows us to calculate that the weight enumerator of the dual code is

$$\begin{aligned} W_{C^\perp}(x,y) &= \frac{1}{2^2}[(x+y)^3 + 3(x+y)(x-y)^2] \\ &= \frac{1}{4}[(x^3 + 3x^2y + 3xy^2 + y^3) + 3(x+y)(x^2 - 2xy + y^2)] \\ &= \frac{1}{4}[x^3 + 3x^2y + 3xy^2 + y^3 + 3x^3 - 6x^2y + 3xy^2 + 3x^2y - 6xy^2 + 3y^3] \\ &= \frac{1}{4}[4x^3 + 4y^3] \\ &= x^3 + y^3. \end{aligned}$$

As discussed, the above weight enumerator for $C^\perp$ was found without using any information of the dual code, indicating that $W_{C^\perp}(x, y)$ is uniquely determined by $W_C(x, y)$ as the MacWilliams Theorem stated. We can also note that the weight enumerator of the dual code found through the MacWilliams Theorem is exactly equal to the one found using the dual code weight enumerator definition in (4.5).

In addition to its polynomial form, the MacWilliams Theorem can also be expressed in many different ways. We will only consider one other form here,

which relies on the Binomial Theorem, and is as follows:

$$A_j^{\perp} = \frac{1}{|C|} \sum_{i=0}^{n} A_i \sum_{l=0}^{j} (-1)^l \binom{i}{l} \binom{n-i}{j-l} \text{ for } 0 \leq j \leq n. \tag{4.15}$$

As a reminder, A_i is the number of codewords in C with weight i and $A_j^{\perp}$ is the number of codewords in $C^{\perp}$ with weight j.

THEOREM 4.6 *Equations* (4.7) *and* (4.15) *are equivalent.*

Proof: From (4.7), we have that

$$W_{C^{\perp}}(x,y) = \frac{1}{|C|} W_C(x+y, x-y) \tag{4.16}$$

$$= \frac{1}{|C|} \sum_{i=0}^{n} A_i (x+y)^{n-i} (x-y)^i. \tag{4.17}$$

Recall that the Binomial Theorem is

$$(x+y)^t = \sum_{k=0}^{t} \binom{t}{k} x^{t-k} y^k.$$

Then, expanding (4.17) using the Binomial Theorem, we obtain

$$W_{C^{\perp}}(x,y) = \frac{1}{|C|} \sum_{i=0}^{n} A_i \sum_{l=0}^{n-i} \binom{n-i}{l} x^{n-i-l} y^l \sum_{l=0}^{i} \binom{i}{l} x^{i-l} (-y)^l \tag{4.18}$$

$$= \frac{1}{|C|} \sum_{i=0}^{n} A_i \sum_{l=0}^{n-i} \binom{n-i}{l} x^{n-i-l} y^l \sum_{l=0}^{i} \binom{i}{l} x^{i-l} (-1)^l y^l. \tag{4.19}$$

By definition, $W_{C^{\perp}}(x,y)$ gives the number of codewords in $C^{\perp}$ for each possible weight from 0 to n. However, in (4.15), $A_j^{\perp}$ gives only the number of codewords in $C^{\perp}$ with weight j. Thus, we want to look only at the j^{th} term of (4.19).

As in the definitions of the weight enumerators, the power of the y term indicates the weight. So, if we want to consider only the j^{th} term, we need to get y^j in (4.19). In order to do this, we want our latter y^l term, located in the third summation, to be y^{j-l} where l in this portion is now $j-l$. Then, when the y terms are multiplied together we would obtain $y^l y^{j-l} = y^j$. Equating the coefficients of the j^{th} term of (4.19), we get that

$$A_j^{\perp} = \frac{1}{|C|} \sum_{i=0}^{n} A_i \sum_{l=0}^{j} (-1)^l \binom{i}{l} \binom{n-i}{j-l}$$

as desired. □

In contrast to the polynomial form of the MacWilliams Theorem, (4.15) gives the number of words in the dual code for one specific weight. For example, if we wanted to find the number of words of weight 1 for the dual code given in (3.7), we calculate $A_1^{\perp}$ using (4.15) and our weight distribution for the associated $[3, 2]$ code C in (4.3). Then,

$$\begin{aligned}
A_3^{\perp} &= \frac{1}{4}\left[A_0 \sum_{l=0}^{3} (-1)^l \binom{0}{l}\binom{3-0}{3-l} + A_2 \sum_{l=0}^{3} (-1)^l \binom{2}{l}\binom{3-2}{3-l}\right] \\
&= \frac{1}{4}\left[\sum_{l=0}^{3} (-1)^l \binom{0}{l}\binom{3}{3-l} + 3\sum_{l=0}^{3} (-1)^l \binom{2}{l}\binom{1}{3-l}\right] \\
&= \frac{1}{4}\left[(1+0+0+0) + 3(0+0+1+0)\right] \\
&= \frac{1}{4}\left[4\right] \\
&= 1.
\end{aligned}$$

So, there is one codeword of weight 1 in the dual code, which is exactly what we found using the polynomial form of the MacWilliams Theorem. Thus, we can conclude that each individual weight of the dual code is also uniquely determined by the weight distribution of the code C.

References

[1] Florence Jessie MacWilliams and Neil James Alexander Sloane The Theory of Error-Correcting Codes. *North-Holland Publishing Company*, 16, 1981.

[2] Judy L. Walker Codes and Curves. *Faculty Publications, Department of Mathematics*, 164, 2000.

[3] Neil James Alexander Sloane Weight Enumerators of Codes *D. Reidel Publishing Company*, 1974.

[4] San Ling and Chaoping Xing Coding Theory A First Course. *Cambridge University Press*, 2004.

[5] David C. Lay, Steven R. Lay, and Judi J. McDonald Linear Algebra and its Applications. *Pearson*, 5th edition, 2015.