Appendix A5: Coding Theory

Coding theory, or the theory of error-correcting codes, concerns how information may best be represented for the purpose of transmission over a noisy channel, or stored in imperfect media, in such a way that a limited number of errors introduced during transmission/reception, or storage/retrieval, may be corrected. To formalize these goals inevitably leads us to notions of finite geometry. Coding theorists use many of the standard constructions of finite geometry in the construction of good codes. In this study, where finite geometry is our primary interest, we shall have more use for the reverse process in which coding theory is used to study finite geometry. For a more comprehensive introduction to coding theory, see e.g. [65].

An alphabet is simply a finite set $A$ of symbols, possibly the Roman alphabet $\{a,b,c,\ldots,z\}$ or the set of 256 ASCII characters; or more typically for our use, the elements of a finite field, especially the binary alphabet $\mathbb{F}_2 = \{0,1\}$ whose elements we call bits. A code of length $n$ over an alphabet $A$ is a subset $C \subseteq A^n$. We refer to $C$ as a $q$-ary code where $q = |A|$. Elements of $A^n$ are called words (or bitstrings if $A = \{0,1\}$), and elements of the subset $C$ are called codewords. The (Hamming) distance between two words $w, w' \in A^n$, denoted $\delta(w, w') \in \{0,1,2,\ldots,n\}$, is the number of coordinates in which they differ. Note that $\delta$ is a metric on $A^n$. The minimum distance of $C$ is

$$d = \min_{w \neq v \in C} \delta(v, w).$$

Denote the closed ball of radius $r$ centred at a vector $w \in \mathbb{F}_q^n$ by

$$B_r(w) = \{v \in \mathbb{F}_q^n : \delta(v, w) \leq r\}.$$ 

A code $C$ is $e$-error correcting if the balls $B_e(w)$ centred at the codewords $w \in C$ are mutually disjoint. In this case a received word $v \in B_e(w)$ is uniquely decoded as $w \in C$, at least in principle. (Actually finding the closest codeword $w \in C$ to a given word $v$ may be difficult in practice.)

A5.1 Proposition. A code $C$ is $e$-error correcting iff its minimum distance is at least $2e + 1$.

Proof. If $C$ is not $e$-error correcting then there exist $w \neq w' \in C$ such that $B_e(w) \cap B_e(w')$ contains some vector $v \in \mathbb{F}_q^n$; but then $\delta(w, w') \leq \delta(w, v) + \delta(v, w') \leq e + e = 2e$. Clearly the converse of this argument also holds.

We say that $C$ is $e$-error detecting if $B_e(w) \cap B_{e-1}(w') = \emptyset$ for all $w \neq w'$ in $C$. This says that if at most $e$ symbols in a transmitted word are altered during transmission, the recipient can be sure that such an error occurred; but cannot in general correct it; see
Example A5.6 below. Clearly every code with minimum distance $d$ is $\left\lceil \frac{d}{2} \right\rceil$-error detecting, and $\left\lceil \frac{d-1}{2} \right\rceil$-error correcting.

Note that

$$|B_r(v)| = 1 + n(q - 1) + \binom{n}{2} (q - 1)^2 + \cdots + \binom{n}{r} (q - 1)^r.$$

An elementary counting argument using this yields

\begin{center}
A5.2 Theorem (Sphere-Packing Bound; Hamming Bound). If $C$ is an $e$-error correcting $q$-ary code of length $n$, then

$$|C| \leq \frac{q^n}{1 + n(q-1) + \binom{n}{2} (q-1)^2 + \cdots + \binom{n}{r} (q-1)^r}.$$\end{center}

A \textbf{perfect code} is one for which equality holds in the Sphere-Packing Bound; in this case the balls $B_e(w)$ centred at codewords $w \in C$ partition the set $A^n$ of all words.

Usually $A = \mathbb{F}_q$ is a finite field, and subspaces $C \leq \mathbb{F}_q^n$ are called \textbf{linear codes}. A linear code of length $n$ and dimension $k$ is called an $[n, k]$-\textbf{code}. The \textbf{(Hamming) weight} of a word (i.e. vector) $v \in \mathbb{F}_q^n$ is the number $wt(v) \in \{0, 1, 2, \ldots, n\}$ of nonzero coordinates of $v$. Note that the Hamming distance between two words $v, w \in \mathbb{F}_q^n$ is given by

$$\delta(v, w) = wt(w - v).$$

It follows that the minimum distance of a linear code $C$ coincides with the \textbf{minimum weight}:

$$d = \min_{\substack{w \neq 0 \\text{in } C}} wt(w).$$

A linear code with minimum weight $d$ is also called an $[n, k, d]$-\textbf{code}.

Let $G$ be a $k \times n$ matrix over $\mathbb{F}_q$ having linearly independent rows (so in particular $k \leq n$). The row space of $G$ is an $[n, k]$-code

$$C = \{xG : x \in \mathbb{F}_q^k\}.$$ We call $G$ a \textbf{generator matrix} for the code $C$. In this case the injective map $\mathbb{F}_q^k \hookrightarrow \mathbb{F}_q^n$, $x \mapsto xG$ gives an encoding of all $q$-ary words of length $k$, as words of length $n$. We say that $C$ has \textbf{information rate} equal to $\frac{k}{n}$. This is the fraction of the transmitted symbols which contain useful information; the remaining symbols may serve to provide error correction.
An \([n, k]\)-code may be specified as the row space of a \(k \times n\) matrix \(G\) as above, or as the right null space of an \((n-k) \times n\) matrix \(H\) having linearly independent rows:

\[
C = \{ y \in \mathbb{F}_q^n : Hy^T = 0 \in \mathbb{F}_q^{n-k} \}.
\]

Such a matrix \(H\) is called a **parity check matrix** for \(C\). Given a received word \(y \in \mathbb{F}_q^n\), we call the vector \(Hy^T \in \mathbb{F}_q^{n-k}\) the **(error) syndrome** of \(y\). The syndrome is useful in error detection and correction: certainly a nonzero syndrome indicates that an error occurred during transmission. To see the utility of syndromes in error correction, see Example A5.8 below.

The fundamental goal of coding theory is to find codes with the following desirable properties:

- High information rate;
- High error-correcting capability (i.e. large minimum distance);
- Efficient encoding and decoding algorithms should be available.

These goals tend to be conflicting in nature; for example one can increase the error-correcting capability, at the cost of reducing the information rate and thereby increasing the cost of transmission, as Theorem A5.2 shows. Another bound which shows this trade-off between minimum distance and information rate is the following.

\begin{center}
A5.3 Theorem (Singleton Bound). If \(C\) is a linear \([n, k, d]\)-code over \(\mathbb{F}_q\) then \\
\[ k \leq n - d + 1. \]
\end{center}

**Proof.** Let \(H\) be an \((n-k) \times n\) parity check matrix for \(C\). Since \(H\) has rank \(n-k\), it has a set of \(n-k\) linearly independent columns; we may suppose that the first \(n-k\) columns of \(H\) are linearly independent, otherwise permute the columns of \(H\) appropriately. (Permuting the coordinates of \(C\) yields once again an \([n, k, d]\)-code.) Now we may assume \(H\) is in reduced row-echelon form; otherwise replace \(H\) by its reduced row-echelon form, and this will not change its right null space. Thus

\[
H = [I_{n-k} \ X]
\]

where \(I\) is the \((n-k) \times (n-k)\) identity matrix, and \(X \in \mathbb{F}_q^{(n-k) \times k}\). It is easy to check that the matrix

\[
G = [-X^T \ I_k]
\]

is a generator matrix for \(C\). Its rows have weight at most \(n-k+1\), so that \(d \leq n-k+1\). \(\square\)

Codes for which the Singleton Bound is attained, are called **MDS codes**. (This is an acronym for **Maximum Distance Separable**, which is such an unfortunate term that
everyone just calls them MDS codes.) Although both bounds A5.2 and A5.3 describe the optimal possible tradeoff between information rate and minimum distance, neither bound implies the other. Thus there exist MDS codes which are not perfect; and there exist perfect codes which are not MDS.

The weight enumerator of a code $C$ of length $n$ is the polynomial

$$A_C(z) = \sum_{v \in C} x^{n-\text{wt}(v)} y^{\text{wt}(v)} = \sum_{0 \leq d \leq n} A_d x^{n-d} y^d$$

where $A_d$ is the number of codewords of weight $d$. The essential information conveyed by the weight enumerator is the list of integer values $A_0, A_1, \ldots, A_n$, known as the weight distribution of $C$. The reason for representing this list of values as a single polynomial is motivated by Theorem A5.4 below. In the case $C$ is a linear code, we define the dual code by

$$C^\perp = \{ w \in \mathbb{F}_q^n : w \cdot v = 0 \text{ for all } v \in C \}$$

where $w \cdot v = wv^T$. Thus the dual of an $[n, k]$-code is an $[n, n-k]$-code. If $C$ has generator matrix $G$ and parity check matrix $H$, then $C^\perp$ has generator matrix $H$ and parity check matrix $G$.

The MacWilliams relations (Theorem A5.4) show that the weight distribution of either of these two codes ($C$ or $C^\perp$) determines the weight distribution of the other.

### A5.4 Theorem (MacWilliams).
Let $C$ be a linear $[n, k]$-code over $\mathbb{F}_q$. Then the weight enumerator of the dual code is given by

$$A_{C^\perp}(x, y) = q^{-k} A_C(x + (q-1)y, x - y).$$

Before proving Theorem A5.4 we consider an example.

### A5.5 Example.
Consider the binary $[5, 2]$-code $C = \{00000, 11100, 10011, 01111\}$, which has weight enumerator

$$A_C(x, y) = x^5 + 2x^2 y^3 + xy^4.$$ 

Its dual is the binary $[5, 3]$-code

$$C^\perp = \{00000, 01100, 00011, 01111, 11010, 11001, 10110, 10101\}$$

whose weight enumerator is

$$A_{C^\perp}(z) = x^5 + 2x^3 y^2 + 4x^2 y^3 + xy^4.$$
The MacWilliams relations are expressed as the identity

\[ A_{C^\perp}(x, y) = \frac{1}{4} A_C(x + y, x - y), \]

i.e.

\[ x^5 + 2x^3y^2 + 4x^2y^3 + xy^4 = \frac{1}{4}[(x+y)^5 + 2(x+y)^2(x-y)^3 + (x+y)(x-y)^4] \]

which may be verified directly. In passing we note that \( C^\perp \cap C = \{00000, 01111\} \). This is typical in that while the subspaces \( C, C^\perp \leq F_q^n \) have complementary dimension, they are not in general complementary subspaces (unlike the situation for inner product spaces).

**Proof of Theorem A5.4.** A character of the additive group of \( F_q \) is a map \( \chi : F_q \to C^\times \) such that \( \chi(a + b) = \chi(a) \chi(b) \) for all \( a, b \in F_q \). (Here \( C^\times \) is the multiplicative group of nonzero complex numbers.) In particular the map \( \chi(a) = 1 \) for all \( a \in F_q \) is the principal character; every other character of \( F_q \) is nonprincipal. Let \( \chi \) be a nonprincipal character of \( F_q \) and observe that

\[ \sum_{a \in F_q} \chi(a) = 0. \]

For all \( u \in F_q^n \) define

\[ g(u) = \sum_{v \in F_q^n} \chi(u \cdot v) x^{n - wt(v)} y^{wt(v)}. \]

Then

\[ \sum_{u \in C} g(u) = \sum_{u \in C} \sum_{v \in F_q^n} \chi(u \cdot v) x^{n - wt(v)} y^{wt(v)} = \sum_{v \in F_q^n} \left( \sum_{u \in C} \chi(u \cdot v) \right) x^{n - wt(v)} y^{wt(v)}. \]

The innermost sum is \( q^k \) for \( v \in C^\perp \) and vanishes otherwise; therefore

\[ \sum_{u \in C} g(u) = q^k \sum_{v \in C^\perp} x^{n - wt(v)} y^{wt(v)} = q^k A_{C^\perp}(x, y). \]

For \( v = (v_1, \ldots, v_n) \in F_q^n \) we have

\[ wt(v) = wt(v_1) + \cdots + wt(v_n) \quad \text{where} \quad wt(v_i) = \begin{cases} 0, & \text{if } v_i = 0; \\ 1, & \text{if } v_i = 1 \end{cases} \]

and so

\[ g(u) = \sum_{v_1, \ldots, v_n \in F_q} \chi(u_1 v_1 + \cdots + u_n v_n) x^{n - wt(v_1) - \cdots - wt(v_n)} y^{wt(v_1) + \cdots + wt(v_n)} \]

\[ = \sum_{v_1, \ldots, v_n \in F_q} \chi(u_1 v_1) x^{1 - wt(v_1)} y^{wt(v_1)} \chi(u_2 v_2) x^{1 - wt(v_2)} y^{wt(v_2)} \cdots \chi(u_n v_n) x^{1 - wt(v_n)} y^{wt(v_n)} \]

\[ = \prod_{1 \leq i \leq n} \sum_{v_i \in F_q} \chi(u_i v_i) x^{1 - wt(v_i)} y^{wt(v_i)}. \]
The innermost sum equals \( x + (q-1)y \) if \( u_i = 0 \), and \( x - y \) if \( u_i \neq 0 \); thus
\[
g(u) = (x + (q-1)y)^{n-\wt(u)}(x - y)^{\wt(u)}.
\]

Summing over \( u \in C \) gives the required result.

Consider the problem of transmitting a single hexadecimal digit, or equivalently, a bitstring of length 4. Such a message may be transmitted as a sequence of four bits, but this scheme will provide no error-correcting capability: if the message 0100 is transmitted, and the third bit is altered due to static so that the word 0110 is received, the error cannot be detected by the receiver. The following examples show how we can do better.

**A5.6 Example: Parity Check Code.** A slightly better scheme would be to add a single parity check bit to every word; thus message words and the corresponding codewords are as listed in the following table. This code detects single errors but corrects no errors.

<table>
<thead>
<tr>
<th>Message Word</th>
<th>Codeword</th>
<th>Message Word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000 0</td>
<td>1000</td>
<td>1000 1</td>
</tr>
<tr>
<td>0001</td>
<td>0001 1</td>
<td>1001</td>
<td>1001 0</td>
</tr>
<tr>
<td>0010</td>
<td>0010 1</td>
<td>1010</td>
<td>1010 0</td>
</tr>
<tr>
<td>0111</td>
<td>0111 0</td>
<td>1111</td>
<td>1111 0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

This code has an information rate of \( \frac{4}{5} = 80\% \), meaning that 4 out of every 5 bits transmitted carry useful information; 1 out of every 5 bits transmitted supply redundancy useful in checking the validity of the information transmitted.

This code has generator matrix and parity check matrix given by
\[
G = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}, \quad H = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}.
\]

**A5.7 Example: 3-Repetition Code.** The idea behind this code is to simply send every bit three times, and to require the receiver to use a simple ‘majority rules’ approach to decoding. This code is not only 1-error detecting, but in fact 1-error correcting. This
advantage comes at the cost: 2 out of every 3 bits transmitted are redundant bits, and the information rate is only $\frac{1}{3} \approx 33\%$.

<table>
<thead>
<tr>
<th>Message Word</th>
<th>Codeword</th>
<th>Message Word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>000 000 000 000</td>
<td>1000</td>
<td>111 000 000 000</td>
</tr>
<tr>
<td>0001</td>
<td>000 000 000 111</td>
<td>1001</td>
<td>111 000 000 111</td>
</tr>
<tr>
<td>0010</td>
<td>000 000 111 000</td>
<td>1010</td>
<td>111 000 111 000</td>
</tr>
<tr>
<td>0011</td>
<td>000 000 111 111</td>
<td>1011</td>
<td>111 000 111 111</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0111</td>
<td>000 111 111 111</td>
<td>1111</td>
<td>111 111 111 111</td>
</tr>
</tbody>
</table>

This code has generator matrix and parity check matrix given by

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

A5.8 Example: The [7, 4, 3] binary Hamming Code. The following scheme is superior to the previous example in that it also allows single bit errors to be corrected, yet it has a substantially higher information rate of $\frac{4}{7} \approx 57\%$. It is in fact a perfect code, although not an MDS code.

<table>
<thead>
<tr>
<th>Message Word</th>
<th>Codeword</th>
<th>Message Word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000 000</td>
<td>1000</td>
<td>1000 011</td>
</tr>
<tr>
<td>0001</td>
<td>0001 111</td>
<td>1001</td>
<td>1001 100</td>
</tr>
<tr>
<td>0010</td>
<td>0010 110</td>
<td>1010</td>
<td>1010 101</td>
</tr>
<tr>
<td>0011</td>
<td>0011 001</td>
<td>1011</td>
<td>1011 010</td>
</tr>
<tr>
<td>0100</td>
<td>0100 101</td>
<td>1100</td>
<td>1100 110</td>
</tr>
<tr>
<td>0101</td>
<td>0101 010</td>
<td>1101</td>
<td>1101 001</td>
</tr>
<tr>
<td>0110</td>
<td>0110 011</td>
<td>1110</td>
<td>1110 000</td>
</tr>
<tr>
<td>0111</td>
<td>0111 100</td>
<td>1111</td>
<td>1111 111</td>
</tr>
</tbody>
</table>

This code has generator matrix and parity check matrix given by

$$G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}, \quad H = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}.$$
Note that the columns of $H$ consist of the numbers one through seven, written in binary. This allows for very easy decoding. Note that a message string $x \in \mathbb{F}_2^4$ is encoded as a codeword $y = xG \in \mathbb{F}_2^7$. Suppose that this word is transmitted and a word $y' \in \mathbb{F}_2^7$ is received. The receiver first computes the syndrome $Hy^r \in \mathbb{F}_2^3$.

- If the syndrome is zero, then $y'$ is a legitimate codeword. We assume no bit errors occurred during transmission (otherwise at least three bit errors occurred, which we deem unlikely). We recover $x \in \mathbb{F}_2^4$ as the first four bits of $y'$.

- If the syndrome is nonzero, then the syndrome is the binary representation of some $i \in \{1, 2, \ldots, 7\}$. First switch the $i$-th bit of $y'$ (i.e. add 1 mod 2 in the $i$-th coordinate) to obtain the unique closest codeword to $y'$. We assume the resulting codeword is actually $y$; so as before, we recover $x \in \mathbb{F}_2^4$ as the first four bits of $y'$.

For example consider the message word $x = 1101$, which is encoded as $y = xG = 1101001$. Suppose this codeword is transmitted but that due to interference, the string $y' = 1111001$ is received. We compute the syndrome

$$Hy^r = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$ 

This is the binary representation of 3, so switch the third bit of $y'$ to recover $y = 1101001$ and thereby we find $x = 1101$. 