

Chapter 1

Basics on error control

1.1 ABC on codes

1.1.1 *Basic notations and terminology*

The basic idea of coding is to introduce redundancy that can be utilized to detect and, for some applications, correct errors that have occurred during a transmission over a channel. Here "transmission" is used in a wide sense, including any process which may corrupt the data, e.g. transmission, storage, etc. The symbols transmitted are from some finite alphabet F . If the alphabet has size q we will sometimes denote it by F_q . We mainly consider channels without memory, that is, a symbol $a \in F$ is transformed to $b \in F$ with some probability $\pi(a, b)$, independent of other symbols transmitted (earlier or later). Since the channel is described by the transition probabilities and a change of alphabet is just a renaming of the symbols, the actual alphabet is not important. However, many code constructions utilize a structure of the alphabet. We will usually assume that the alphabet of size q is the set Z_q of integers modulo q . When q is a prime power, we will sometimes use the finite field $GF(q)$ as alphabet. The main reason is that vector spaces over finite fields are important codes; they are called linear codes.

As usual, F^n denotes the set of n -tuples (a_1, a_2, \dots, a_n) where $a_i \in F$. The n -tuples will also be called *vectors*.

Suppose that we have a set \mathcal{M} of M possible messages that may be sent. An $(n, M; q)$ code is a subset of F^n containing M vectors. An *encoding* is a one-to-one function from \mathcal{M} to the code. The vectors of the code are called *code words*.

1.1.2 Hamming weight and distance

The *Hamming weight* $w_H(\mathbf{x})$ of a vector \mathbf{x} is the number of non-zero positions in \mathbf{x} , that is

$$w_H(\mathbf{x}) = \#\{i \mid 1 \leq i \leq n \text{ and } x_i \neq 0\}.$$

The *Hamming distance* $d_H(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x}, \mathbf{y} \in F_q^n$ is the number of positions where they differ, that is

$$d_H(\mathbf{x}, \mathbf{y}) = \#\{i \mid 1 \leq i \leq n \text{ and } x_i \neq y_i\}.$$

If a vector \mathbf{x} was transmitted and e errors occurred during transmission, then the received vector \mathbf{y} differs from \mathbf{x} in e positions, that is $d_H(\mathbf{x}, \mathbf{y}) = e$.

Clearly,

$$d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} - \mathbf{y}).$$

For an $(n, M; q)$ code C , define the *minimum distance* by

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

and let

$$d(n, M; q) = \max\{d(C) \mid C \text{ is an } (n, M; q) \text{ code}\}.$$

Sometimes we include $d = d(C)$ in the notation for a code and write $(n, M, d; q)$ and $[n, k, d; q]$. The *rate* of a code $C \subset F_q^n$ is

$$R = \frac{\log_q \#C}{n}.$$

Define

$$\delta(n, R; q) = \frac{d(n, \lceil q^{Rn} \rceil; q)}{n},$$

and

$$\delta(R; q) = \limsup_{n \rightarrow \infty} \delta(n, R; q).$$

1.1.3 Support of a set of vectors

For $\mathbf{x} = (x_1, x_2, \dots, x_n), \mathbf{y} = (y_1, y_2, \dots, y_n) \in F^n$, we define the *support* of the pair (\mathbf{x}, \mathbf{y}) by

$$\chi(\mathbf{x}, \mathbf{y}) = \{i \in \{1, 2, \dots, n\} \mid x_i \neq y_i\}.$$

Note that

$$\#\chi(\mathbf{x}, \mathbf{y}) = d_H(\mathbf{x}, \mathbf{y}).$$

For a single vector $\mathbf{x} \in F^n$, the support is defined by $\chi(\mathbf{x}) = \chi(\mathbf{x}, \mathbf{0})$.

For a set $S \subseteq F^n$, we define its *support* by

$$\chi(S) = \bigcup_{\mathbf{x}, \mathbf{y} \in S} \chi(\mathbf{x}, \mathbf{y}).$$

In particular, $\chi(S)$ is the set of positions where not all vectors in S are equal.

1.1.4 Extending vectors

Let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in F^n$, $\mathbf{y} = (y_1, y_2, \dots, y_m) \in F^m$ and $u \in F$. Then

$$u\mathbf{x} = (ux_1, ux_2, \dots, ux_n),$$

$$(\mathbf{x}|u) = (x_1, x_2, \dots, x_n, u),$$

$$(\mathbf{x}|\mathbf{y}) = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m).$$

The last two operations are called concatenation. For a subset S of F^n ,

$$uS = \{u\mathbf{x} \mid \mathbf{x} \in S\},$$

$$(S|u) = \{(\mathbf{x}|u) \mid \mathbf{x} \in S\}.$$

1.1.5 Ordering

Let some ordering \leq of F be given. We extend this to a partial ordering of F^n as follows:

$$(x_1, x_2, \dots, x_n) \leq (y_1, y_2, \dots, y_n) \text{ if } x_i \leq y_i \text{ for } 1 \leq i \leq n.$$

For Z_q we use the natural ordering $0 < 1 < 2 \dots < q - 1$.

1.1.6 Entropy

The base q (or q -ary) entropy function $H_q(z)$ is defined by

$$H_q(z) = -z \log_q \left(\frac{z}{q-1} \right) - (1-z) \log_q(1-z)$$

for $0 \leq z \leq 1$. $H_q(z)$ is an increasing function on $\left[0, \frac{q-1}{q}\right]$, $H_q(0) = 0$, and $H_q\left(\frac{q-1}{q}\right) = 1$. Define $\rho(z) = \rho_q(z)$ on $[0, 1]$ by $\rho_b(z) \in \left[0, \frac{q-1}{q}\right]$ and

$$H_b(\rho_b(z)) = 1 - z.$$

1.1.7 Systematic codes

An $(n, q^k; q)$ code C is called *systematic* if it has the form

$$C = \{(\mathbf{x}|f(\mathbf{x})) \mid \mathbf{x} \in F_q^k\}$$

where f is a mapping from F_q^k to F_q^{n-k} . Here $(\mathbf{x}|f(\mathbf{x}))$ denotes the concatenation of \mathbf{x} and $f(\mathbf{x})$.

1.1.8 Equivalent codes

Two $(n, M; q)$ codes C_1, C_2 are *equivalent* if C_2 can be obtained from C_1 by permuting the positions of all code words by the same permutation. We note that equivalent codes have the same distance distribution, and in particular the same minimum distance.

1.1.9 New codes from old

There are a number of ways to construct new codes from one or more old ones. We will describe some of these briefly. In a later section we will discuss how the error detecting capability of the new codes are related to the error detecting capability of the old ones.

Extending a code

Consider an $(n, M; q)$ code C . Let $\mathbf{b} = (b_1, b_2, \dots, b_n) \in F_q^n$. Let C^{ex} be the $(n + 1, M; q)$ code

$$C^{\text{ex}} = \left\{ (a_1, a_2, \dots, a_n, - \sum_{i=1}^n a_i b_i) \mid (a_1, a_2, \dots, a_n) \in C \right\}.$$

Note that this construction depends on the algebraic structure of the alphabet F_p (addition and multiplication are used to define the last term). For example, let $n = 2$, $\mathbf{b} = (1, 1)$, and $\mathbf{a} = (1, 1)$. If the alphabet is $GF(4)$, then $a_1 b_1 + a_2 b_2 = 0$, but if the alphabet is Z_4 , then $a_1 b_1 + a_2 b_2 = 2 \neq 0$.

Puncturing a code

Consider an $(n, M; q)$ code. *Puncturing* is to remove the first position from each code word (puncturing can also be done in any other position). This produces a code C^{p} of length $n - 1$. If two code words in C are identical, except in the first position, then the punctured code words are the same. Hence the size of C^{p} may be less than M . On the other hand, any code word $\mathbf{c} \in C^{\text{p}}$ is obtained from a vector $(a|\mathbf{c})$ where $a \in F_q$. Hence, the size of C^{p} is at least M/q . The minimum distance may decrease by one. Clearly, the operation of puncturing may be repeated.

Shortening a code

Consider an $(n, M; q)$ code C with the first position in its support. *Shortening* (by the first position) we obtain the $(n - 1, M'; q)$ code

$$C^s = \left\{ \mathbf{x} \in F^{n-1} \mid (0|\mathbf{x}) \in C \right\},$$

that is, we take the set of all code words of C with 0 in the first position and remove that position. More general, we can shorten by any position in the support of the code.

We note that shortening will not decrease the minimum distance; however it may increase it. In the extreme case, when there are no code words in C with 0 in the first position, C^s is empty.

Zero-sum subcodes of a code

Consider an $(n, M; q)$ code C . The *zero-sum subcode* C^{zs} is the code

$$C^{zs} = \left\{ (a_1, a_2, \dots, a_n) \in C \mid \sum_{i=1}^n a_i = 0 \right\}.$$

Also this construction depends on the algebraic structure of the alphabet.

In the binary case, $\sum_{i=1}^n a_i = 0$ if and only if $w_H(\mathbf{a})$ is even, and C^{zs} is then called the *even-weight subcode*.

1.1.10 Cyclic codes

A code $C \subseteq F^n$ is called *cyclic* if

$$(a_{n-1}, a_{n-2}, \dots, a_0) \in C \text{ implies that } (a_{n-2}, a_{n-3}, \dots, a_0, a_{n-1}) \in C.$$

Our reason for the special way of indexing the elements is that we want to associate a polynomial in the variable z with each n -tuple as follows:

$$\mathbf{a} = (a_{n-1}, a_{n-2}, \dots, a_0) \leftrightarrow a(z) = a_{n-1}z^{n-1} + a_{n-2}z^{n-2} \dots + a_0.$$

This correspondence has the following property (it is an isomorphism): if $\mathbf{a}, \mathbf{b} \in F^n$ and $c \in F$, then

$$\mathbf{a} + \mathbf{b} \leftrightarrow a(z) + b(z),$$

$$c\mathbf{a} \leftrightarrow ca(z).$$

In particular, any code may be represented as a set of polynomials. Moreover, the polynomial corresponding to $(a_{n-2}, a_{n-3}, \dots, a_0, a_{n-1})$ is

$$a_{n-1} + \sum_{i=0}^{n-2} a_i z^{i+1} = za(z) - a_{n-1}(z^n - 1) \equiv za(z) \pmod{z^n - 1}.$$

1.2 Linear codes

An $[n, k; q]$ linear code is a k -dimensional subspace of $GF(q)^n$. This is in particular an $(n, q^k; q)$ code. Vector spaces can be represented in various ways and different representations are used in different situations.

1.2.1 Generator and check matrices for linear codes

Suppose that $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k\}$ is a basis for C . Then C is the set of all possible linear combinations of these vectors. Let G be the $k \times n$ matrix whose k rows are $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k$. Then

$$C = \{\mathbf{x}G \mid \mathbf{x} \in GF(q)^k\}.$$

We call G a *generator matrix* for C . A natural *encoding* $GF(q)^k \rightarrow GF(q)^n$ is given by

$$\mathbf{x} \mapsto \mathbf{x}G.$$

If $T : GF(q)^k \rightarrow GF(q)^k$ is a linear invertible transformation, then TG is also a generator matrix. The effect is just a change of basis.

The *inner product* of two vectors $\mathbf{x}, \mathbf{y} \in GF(q)^n$ is defined by

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{xy}^t = \sum_{i=1}^n x_i y_i,$$

where \mathbf{y}^t is the transposed of \mathbf{y} . For a linear $[n, k; q]$, the dual code is the $[n, n - k; q]$ code

$$C^\perp = \{\mathbf{x} \in GF(q)^n \mid \mathbf{xc}^t = 0 \text{ for all } \mathbf{c} \in C\}.$$

If H is a generator matrix for C^\perp , then

$$C = \{\mathbf{x} \in GF(q)^n \mid \mathbf{xH}^t = 0\},$$

where H^t is the transposed of H . H is known as a (*parity*) *check matrix* for C . Note that $GH^t = 0$ and that any $(n - k) \times n$ matrix H of rank $n - k$ such that $GH^t = 0$ is a check matrix.

1.2.2 The simplex codes and the Hamming codes

Before we go on, we define two classes of codes, partly because they are important in their own right, partly because they are used in other constructions.

Let Γ_k be a $k \times \frac{q^k-1}{q-1}$ matrix over $GF(q)$ such that

- (i) all columns of Γ_k are non-zero,
- (ii) if $\mathbf{x} \neq \mathbf{y}$ are columns, then $\mathbf{x} \neq j\mathbf{y}$ for all $j \in GF(q)$.

The matrix Γ_k generates a $\left[\frac{q^k-1}{q-1}, k, q^{k-1}; q \right]$ code S_k whose non-zero code words all have weight q^{k-1} . It is known as the *Simplex code*. The dual code is an $\left[\frac{q^k-1}{q-1}, \frac{q^k-1}{q-1} - k, 3; q \right]$ code known as the *Hamming code*.

1.2.3 Equivalent and systematic linear codes

Let C_1 be an $[n, k; q]$ code and let

$$C_2 = \{ \mathbf{x}Q\Pi \mid \mathbf{x} \in C_1 \}$$

where Q is a non-singular diagonal $n \times n$ matrix and Π is an $n \times n$ permutation matrix. If G is a generator matrix for C_1 , then $GQ\Pi$ is a generator matrix for C_2 .

Let G be a $k \times n$ generator matrix for some linear code C . By suitable row operations this can be brought into reduced echelon form. This matrix will generate the same code. A suitable permutation of the columns will give a matrix of the form $(I_k|P)$ which generates a systematic code. Here I_k is the identity matrix and P is some $k \times (n-k)$ matrix. Therefore, any linear code is equivalent to a systematic linear code. Since

$$(I_k|P)(-P^t|I_{n-k})^t = -P + P = 0,$$

$H = (-P^t|I_{n-k})$ is a check matrix for C .

1.2.4 New linear codes from old

Extending a linear code

If C is a linear code, then C^{ex} is also linear. Moreover, if H is a check matrix for C , then a check matrix for C^{ex} (where C is extended by \mathbf{b}) is

$$\begin{pmatrix} H & \mathbf{0}^t \\ \mathbf{b} & 1 \end{pmatrix}.$$

In particular, in the binary case, if $b_1 = b_2 = \dots = b_n = 1$, we have extended the code with a parity check. The code $(GF(2)^n)^{\text{ex}}$ is known as the *single parity check code* or just the parity check code.

Shortening a linear code

Shortening a linear code gives a new linear code. If $G = (I_k|P)$ generates a systematic linear code and the code is shortened by the first position, then a generator matrix for the shortened code is obtained by removing the first row and the first column of G .

Puncturing a linear code

Consider puncturing an $[n, k, d; q]$ code C . If the position punctured is not in the support of C , then C^p is an $[n - 1, k, d; q]$ code. If the position punctured is in the support of C , then C^p is an $[n - 1, k - 1, d'; q]$ code. If $d > 1$, then $d' = d$ or $d' = d - 1$. If $d = 1$, then d' can be arbitrary large. For example, if C is the $[n, 2, 1; 2]$ code generated by $(1, 0, 0, \dots, 0)$ and $(1, 1, 1, \dots, 1)$, and we puncture the first position, the resulting code is a $[n - 1, 1, n - 1; 2]$ code.

*The *-operation for linear codes*

Let C be an $[n, k; q]$ code over $GF(q)$. Let C^* denote the $\left[n + \frac{q^k - 1}{q - 1}, k; q\right]$ code obtained from C by extending each code word in C by a distinct code word from the simplex code S_k . We remark that the construction is not unique since there are many ways to choose the code words from S_k . However, for error detection they are equally good (we will return to this later).

We also consider iterations of the *-operation. We define C^{r*} by

$$\begin{aligned} C^{0*} &= C, \\ C^{(r+1)*} &= (C^{r*})^*. \end{aligned}$$

Product codes

Let C_1 be an $[n_1, k_1, d_1; q]$ code and C_2 an $[n_2, k_2, d_2; q]$ code. The *product code* is the $[n_1 n_2, k_1 k_2, d_1 d_2; q]$ code C whose code words are usually written as an $n_1 \times n_2$ array; C is the set of all such arrays where all rows belong to C_1 and all columns to C_2 .

Tensor product codes

Let C_1 be an $[n_1, k_1; q]$ code with parity check matrix

$$H_1 = (h_{ij}^{[1]})_{1 \leq i \leq n_1 - k_1, 1 \leq j \leq n_1}$$

and C_2 an $[n_2, k_2; q]$ code with parity check matrix

$$H_2 = (h_{ij}^{[2]})_{1 \leq i \leq n_2 - k_2, 1 \leq j \leq n_2}.$$

The tensor product code is the $[n_1 n_2, n_1 k_2 + n_2 k_1 - k_1 k_2; q]$ code with parity matrix $H = (h_{ij})$ which is the tensor product of H_1 and H_2 , that is

$$h_{i_1(n_2 - k_2) + i_2, j_1 n_2 + j_2} = h_{i_1, j_1}^{[1]} h_{i_2, j_2}^{[2]}.$$

Repeated codes

Let C be an $(n, M; q)$ code and let r be a positive integer. The r times repeated code, C^r is the code

$$C^r = \{(\mathbf{c}_1 | \mathbf{c}_2 | \cdots | \mathbf{c}_r) \mid \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r \in C\},$$

that is, the Cartesian product of r copies of C . This is an $(rn, Mr; q)$ code with the same minimum distance as C .

Concatenated codes

Codes can be concatenated in various ways. One such construction that has been proposed for a combined error correction and detection is the following.

Let C_1 be an $[N, K; q]$ code and C_2 an $[n, k; q]$ code, where $N = mk$ for some integer m . The encoding is done as follows: K information symbols are encoded into N symbols using code C_1 . These $N = mk$ are split into m blocks with k symbols in each block. Then each block is encoded into n symbols using code C_2 . The concatenated code is an $[mn, K; q]$ code. If G_1 and G_2 are generator matrices for C_1 and C_2 respectively, then a generator matrix for the combined code is the following.

$$G_1 \begin{pmatrix} G_2 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_2 \end{pmatrix}.$$

The construction above can be generalized in various ways. One generalization that is used in several practical systems combines a convolutional code for error correction and a block code (e.g. an CRC code) for error detection.

1.2.5 Cyclic linear and shortened cyclic linear codes

Many important codes are cyclic linear codes or shortened cyclic linear codes. One reason that cyclic codes are used is that they have more algebraic structure than linear codes in general, and this structure can be used both in the analysis of the codes and in the design of efficient encoders and decoders for error correction. For example, the roots of the polynomial $g(z)$, given by the theorem below, give information on the minimum distance of the code. Hamming codes is one class of cyclic codes and shortened Hamming codes and their cosets are used in several standards for data transmission where error detection is important. This is our main reason for introducing them in this text.

Theorem 1.1. *Let C be a cyclic $[n, k; q]$ code. Then there exists a monic polynomial $g(z)$ of degree $n - k$ such that*

$$C = \{v(z)g(z) \mid \deg(v(z)) < k\}.$$

Proof. Let $g(z)$ be the monic polynomial in C of smallest positive degree, say degree m . Then $z^i g(z) \in C$ for $0 \leq i < n - m$. Let $a(z)$ be any non-zero polynomial in C , of degree s , say; $m \leq s < n$. Then there exist elements $c_{s-m}, c_{s-m-1}, \dots, c_0 \in GF(q)$ such that

$$r(z) = a(z) - \sum_{i=0}^{s-m} c_i z^i g(z)$$

has degree less than m (this can easily be shown by induction on s). Since C is a linear code, $r(z) \in C$. Moreover, there exists a $c \in GF(q)$ such that $cr(z)$ is monic, and the minimality of the degree of $g(z)$ implies that $r(z)$ is identically zero. Hence $a(z) = v(z)g(z)$ where $v(z) = \sum_{i=0}^{s-m} c_i z^i$. In particular, the set

$$\{g(z), zg(z), \dots, z^{n-1-m}g(z)\}$$

of $n - m$ polynomials is a basis for C and so $n - m = k$, that is

$$k = n - m. \quad \square$$

The polynomial $g(z)$ is called the *generator polynomial* of C .

If $g(1) \neq 0$, then the code generated by $(z-1)g(z)$ is an $[n+1, k; q]$ code. It is the code C^{ex} obtained from C extending using the vector $\mathbf{1} = (11 \cdots 1)$, that is

$$\left\{ (a_1, a_2, \dots, a_n, -\sum_{i=1}^n a_i) \mid (a_1, a_2, \dots, a_n) \in C \right\}.$$

Encoding using a cyclic code is usually done in one of two ways. Let $\mathbf{v} = (v_{k-1}, v_{k-2}, \dots, v_0) \in GF(q)^k$ be the information to be encoded. The first, and direct way of encoding, is to encode into $v(z)g(z)$. On the other hand, the code is systematic, but this encoding is not. The other way of encoding is to encode \mathbf{v} into the polynomial in C "closest" to $z^{n-k}v(z)$. More precisely, there is a unique $a(z)$ of degree less than k such that

$$-r(z) = z^{n-k}v(z) - a(z)g(z)$$

has degree less than $n - k$, and we encode into

$$a(z)g(z) = z^{n-k}v(z) + r(z).$$

The corresponding code word has the form $(\mathbf{v}|\mathbf{r})$, where $\mathbf{r} \in GF(q)^{n-k}$.

Theorem 1.2. *Let C be a cyclic $[n, k; q]$ code with generator polynomial $g(z)$. Then $g(z)$ divides $z^n - 1$, that is, there exists a monic polynomial $h(z)$ of degree k such that*

$$g(z)h(z) = z^n - 1.$$

Moreover, the polynomial

$$\tilde{h}(z) = -g(0)z^k h\left(\frac{1}{z}\right)$$

is the generator polynomial of C^\perp .

Proof. There exist unique polynomials $h(z)$ and $r(z)$ such that

$$z^n - 1 = g(z)h(z) + r(z)$$

and $\deg(r(z)) < n - k$. In particular $r(z) \equiv h(z)g(z) \pmod{z^n - 1}$ and so $r(z) \in C$. The minimality of the degree of $g(z)$ implies that $r(z) \equiv 0$.

Let $g(z) = \sum_{i=0}^{n-k} g_i z^i$ and $h(z) = \sum_{i=0}^k h_i z^i$. Then

$$\sum_i g_{l-i} h_i = \begin{cases} -1 & \text{if } l = 0, \\ 0 & \text{if } 0 < l < n, \\ 1 & \text{if } l = n. \end{cases}$$

Further, $\tilde{h}(z) = -g_0 \sum_{i=0}^k h_{k-i} z^i$. Since $-g_0 h_0 = 1$, $\tilde{h}(z)$ is monic. Let

$$\mathbf{v} = (0, 0, \dots, 0, g_{n-k}, \dots, g_0), \quad \mathbf{u} = (0, 0, \dots, 0, h_0, \dots, h_k),$$

and let $\mathbf{v}^l, \mathbf{u}^l$ be the vectors l times cyclicly shifted, that is

$$\mathbf{u}^l = (h_{k-l+1}, h_{k-l+2}, \dots, h_k, 0, \dots, 0, h_0, h_1, \dots, h_{k-l}),$$

and \mathbf{v}^l similarly. First, we see that

$$\mathbf{v} \cdot \mathbf{u}^l = \sum_{i=0}^k g_{k+l-i} h_i = 0$$

for $-k < l < n - k$. Hence,

$$\mathbf{v}^m \cdot \mathbf{u}^l = \mathbf{v} \cdot \mathbf{u}^{l-m} = 0$$

for $0 \leq m < k$ and $0 \leq l < n - k$; that is, each basis vector for C is orthogonal to each basis vector in the code \tilde{C} generated by $\tilde{h}(z)$, and so $\tilde{C} = C^\perp$. \square

The polynomial $g(z)$ of degree m is called *primitive* if the least positive n such that $g(z)$ divides $z^n - 1$ is $n = (q^m - 1)/(q - 1)$. The cyclic code C generated by a primitive $g(z)$ of degree m is a $\left[\frac{q^m - 1}{q - 1}, \frac{q^m - 1}{q - 1} - m; q \right]$ *Hamming code*.

The code obtained by shortening the cyclic $[n, k; q]$ code C m times is the $[n - m, k - m; q]$ code

$$\{v(z)g(z) \mid \deg(v(z)) < k'\} \tag{1.1}$$

where $k' = k - m$. Note that (1.1) defines an $[n - k + k', k'; q]$ code for all $k' > 0$, not only for $k' \leq k$. These codes are also known as *cyclic redundancy-check* (CRC) codes. The dual of an $[n - k + k', k'; q]$ code C generated by $g(z) = \sum_{i=0}^{n-k} g_i z^i$ where $g_{n-k} = 1$ can be described as a systematic code as follows: The information sequence (a_{n-k-1}, \dots, a_0) is encoded into the sequence $(a_{n-k-1+k'}, \dots, a_0)$ where

$$a_j = - \sum_{i=0}^{n-k-1} g_i a_{j-n+k+i}.$$

This follows from the fact that

$$(a_{n-k-1+k'}, a_{n-k-2+k'}, \dots, a_0) \cdot (0, 0, \dots, 0, g_{n-k}, \dots, g_0, \overbrace{0, \dots, 0}^i) = 0$$

for $0 \leq i \leq k'$ by definition and that $(0, 0, \dots, 0, g_{n-k}, \dots, g_0, \overbrace{0, \dots, 0}^i)$ where $0 \leq i \leq k'$ is a basis for C .

A number of binary CRC codes have been selected as international standards for error detection in various contexts. We will return to a more detailed discussion of these and other binary CRC codes in Section 3.5.

1.3 Distance distribution of codes

1.3.1 Definition of distance distribution

Let C be an $(n, M; q)$ code. Let

$$A_i = A_i(C) = \frac{1}{M} \#\{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C \text{ and } d_H(\mathbf{x}, \mathbf{y}) = i\}.$$

The sequence A_0, A_1, \dots, A_n is known as the *distance distribution* of C and

$$A_C(z) = \sum_{i=0}^n A_i z^i$$

is the *distance distribution function* of C .

We will give a couple of alternative expressions for the distance distribution function that will be useful in the study of the probability of undetected error for error detecting codes.

1.3.2 The MacWilliams transform

Let C be an $(n, M; q)$ code. The *MacWilliams transform* of $A_C(z)$ is defined by

$$A_C^\perp(z) = \frac{1}{M} (1 + (q-1)z)^n A_C \left(\frac{1-z}{1+(q-1)z} \right). \quad (1.2)$$

Clearly, $A_C^\perp(z)$ is a polynomial in z and we denote the coefficients of $A_C^\perp(z)$ by $A_i^\perp = A_i^\perp(C)$, that is,

$$A_C^\perp(z) = \sum_{i=0}^n A_i^\perp z^i.$$

In particular, $A_0^\perp = 1$.

The reason we use the notation $A_C^\perp(z)$ is that if C is a linear code, then $A_C^\perp(z) = A_{C^\perp}(z)$ as we will show below (Theorem 1.14). However, $A_C^\perp(z)$ is sometimes useful even if C is not linear. The least $i > 0$ such that $A_i^\perp(C) \neq 0$ is known as the *dual distance* $d^\perp(C)$.

Substituting $\frac{1-z}{1+(q-1)z}$ for z in the definition of $A_C^\perp(z)$ we get the following inverse relation.

Lemma 1.1.

$$A_C(z) = \frac{M}{q^n} (1 + (q-1)z)^n A_C^\perp \left(\frac{1-z}{1+(q-1)z} \right). \quad (1.3)$$

Differentiating the polynomial (1.3) s times and putting $z = 1$ we get the following relations which are known as the *Pless identities*.

Theorem 1.3. *Let C be an $(n, M; q)$ code and $s \geq 0$. Then*

$$\sum_{i=0}^n A_i \binom{i}{s} = \frac{M}{q^s} \sum_{j=0}^s A_j^\perp (-1)^j (q-1)^{s-j} \binom{n-j}{s-j}.$$

In particular, if $s < d^\perp$, then

$$\sum_{i=0}^n A_i \binom{i}{s} = \frac{M(q-1)^s}{q^s} \binom{n}{s}.$$

From (1.2) we similarly get the following relation.

Theorem 1.4. *Let C be an $(n, M; q)$ code and $s \geq 0$. Then*

$$\sum_{i=0}^n A_i^\perp \binom{i}{s} = \frac{q^{n-s}}{M} \sum_{j=0}^s A_j (-1)^j (q-1)^{s-j} \binom{n-j}{s-j}.$$

In particular, if $s < d$, then

$$\sum_{i=0}^n A_i^\perp \binom{i}{s} = \frac{q^{n-s}(q-1)^s}{M} \binom{n}{s}.$$

Two important relations are the following.

Theorem 1.5. *Let C be an $(n, M; q)$ code over Z_q and let $\zeta = e^{2\pi\sqrt{-1}/q}$. Then*

$$A_i^\perp(C) = \frac{1}{M^2} \sum_{\substack{\mathbf{u} \in Z_q^n \\ w_H(\mathbf{u})=i}} \left| \sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} \right|^2$$

for $0 \leq i \leq n$.

Note that $\zeta^q = 1$, but $\zeta^j \neq 1$ for $0 < j < q$. Before we prove Theorem 1.5, we first give two lemmas.

Lemma 1.2. *Let $v \in Z_q$. Then*

$$\sum_{u \in Z_q} \zeta^{uv} x^{w_H(u)} = \begin{cases} 1 + (q-1)x & \text{if } v = 0, \\ 1 - x & \text{if } v \neq 0. \end{cases}$$

Proof. We have

$$\sum_{u \in Z_q} \zeta^{uv} x^{w_H(u)} = 1 + x \sum_{u=1}^{q-1} \zeta^{uv}.$$

If $v = 0$, the sum is clearly $1 + x(q - 1)$. If $v \neq 0$, then

$$\sum_{u=1}^{q-1} \zeta^{uv} = -1 + \sum_{u=0}^{q-1} (\zeta^v)^u = -1 + \frac{1 - \zeta^{vq}}{1 - \zeta^v} = -1. \quad \square$$

Lemma 1.3. Let $\mathbf{v} \in Z_q$. Then

$$\sum_{\mathbf{u} \in Z_q^n} \zeta^{\mathbf{u} \cdot \mathbf{v}} x^{w_H(\mathbf{u})} = (1 - x)^{w_H(\mathbf{v})} (1 + (q - 1)x)^{n - w_H(\mathbf{v})}.$$

Proof. From the previous lemma we get

$$\begin{aligned} & \sum_{\mathbf{u} \in Z_q^n} \zeta^{\mathbf{u} \cdot \mathbf{v}} x^{w_H(\mathbf{u})} \\ &= \sum_{u_1 \in Z_q} \zeta^{u_1 v_1} x^{w_H(u_1)} \sum_{u_2 \in Z_q} \zeta^{u_2 v_2} x^{w_H(u_2)} \dots \sum_{u_n \in Z_q} \zeta^{u_n v_n} x^{w_H(u_n)} \\ &= (1 - x)^{w_H(\mathbf{v})} (1 + (q - 1)x)^{n - w_H(\mathbf{v})}. \end{aligned} \quad \square$$

We can now prove Theorem 1.5.

Proof. Since $d_H(\mathbf{c}, \mathbf{c}') = w_H(\mathbf{c} - \mathbf{c}')$, Lemma 1.3 gives

$$\begin{aligned} \sum_{i=0}^n A_i^\perp x^i &= \frac{1}{M} \sum_{i=0}^n A_i (1 - x)^i (1 + (q - 1)x)^{n-i} \\ &= \frac{1}{M^2} \sum_{\mathbf{c} \in C} \sum_{\mathbf{c}' \in C} (1 - x)^{d_H(\mathbf{c}, \mathbf{c}')} (1 + (q - 1)x)^{n - d_H(\mathbf{c}, \mathbf{c}')} \\ &= \frac{1}{M^2} \sum_{\mathbf{c} \in C} \sum_{\mathbf{c}' \in C} \sum_{\mathbf{u} \in Z_q^n} \zeta^{\mathbf{u} \cdot (\mathbf{c} - \mathbf{c}')} x^{w_H(\mathbf{u})} \\ &= \frac{1}{M^2} \sum_{\mathbf{u} \in Z_q^n} x^{w_H(\mathbf{u})} \sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} \sum_{\mathbf{c}' \in C} \zeta^{-\mathbf{u} \cdot \mathbf{c}'}. \end{aligned}$$

Observing that

$$\sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} \sum_{\mathbf{c}' \in C} \zeta^{-\mathbf{u} \cdot \mathbf{c}'} = \left| \sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} \right|^2,$$

the theorem follows. □

When the alphabet is $GF(q)$, there is a similar expression for $A_i^\perp(C)$. Let $q = p^r$, where p is a prime. The *trace function* from $GF(q)$ to $GF(p)$ is defined by

$$\text{Tr}(a) = \sum_{i=0}^{r-1} a^{p^i}.$$

One can show that $\text{Tr}(a) \in GF(p)$ for all $a \in GF(q)$, and that $\text{Tr}(a+b) = \text{Tr}(a) + \text{Tr}(b)$.

Theorem 1.6. *Let $q = p^r$ where p is a prime. Let C be an $(n, M; q)$ code over $GF(q)$ and let $\zeta = e^{2\pi\sqrt{-1}/p}$. Then*

$$A_i^\perp(C) = \frac{1}{M^2} \sum_{\substack{\mathbf{u} \in GF(q)^n \\ w_H(\mathbf{u})=i}} \left| \sum_{\mathbf{c} \in C} \zeta^{\text{Tr}(\mathbf{u} \cdot \mathbf{c})} \right|^2$$

for $0 \leq i \leq n$.

The proof is similar to the proof of Theorem 1.5.

Corollary 1.1. *Let C be an $(n, M; q)$ code (over Z_q or over $GF(q)$). Then*

$$A_i^\perp(C) \geq 0 \text{ for } 0 \leq i \leq n.$$

1.3.3 Binomial moment

We have

$$\begin{aligned} \sum_{j=1}^n A_j x^j &= \sum_{j=1}^n A_j x^j (x+1-x)^{n-j} \\ &= \sum_{j=1}^n A_j x^j \sum_{l=0}^{n-j} \binom{n-j}{l} x^l (1-x)^{n-j-l} \\ &= \sum_{i=1}^n x^i (1-x)^{n-i} \sum_{j=1}^i A_j \binom{n-j}{i-j}. \end{aligned} \quad (1.4)$$

The *binomial moment* is defined by

$$A_i^\diamond(C) = \sum_{j=1}^i A_j(C) \binom{n-j}{n-i}$$

for $1 \leq i \leq n$.

The relation (1.4) then can be expressed as follows:

Theorem 1.7. Let C be an $(n, M; q)$ code. Then

$$A_C(x) = 1 + \sum_{i=1}^n A_i^\diamond(C) x^i (1-x)^{n-i}.$$

We note that the A_i can be expressed in terms of the A_j^\diamond .

Theorem 1.8.

$$A_i(C) = \sum_{j=1}^i (-1)^{j-i} A_j^\diamond(C) \binom{n-j}{n-i}$$

for $1 \leq i \leq n$.

Proof.

$$\begin{aligned} \sum_{j=1}^i (-1)^{j-i} A_j^\diamond(C) \binom{n-j}{n-i} &= \sum_{j=1}^i (-1)^{j-i} \binom{n-j}{n-i} \sum_{k=1}^j A_k(C) \binom{n-k}{n-j} \\ &= \sum_{k=1}^i A_k(C) \sum_{j=k}^i (-1)^{j-i} \binom{n-j}{n-i} \binom{n-k}{n-j} \\ &= \sum_{k=1}^i A_k(C) \sum_{j=k}^i (-1)^{j-i} \binom{n-k}{n-i} \binom{i-k}{i-j} \\ &= \sum_{k=1}^i A_k(C) \binom{n-k}{n-i} (-1)^{i-k} \sum_{j=k}^i (-1)^{j-k} \binom{i-k}{j-k} \\ &= \sum_{k=1}^i A_k(C) \binom{n-k}{n-i} (-1+1)^{i-k} \\ &= A_i(C). \end{aligned}$$

□

We can also express A_i^\diamond in terms of the A_j^\perp . We have

$$\begin{aligned}
 A_C(x) - 1 &= \frac{M}{q^n} \sum_{j=0}^n A_j^\perp (1-x)^j (1+(q-1)x)^{n-j} - 1 \\
 &= \frac{M}{q^n} \sum_{j=0}^n A_j^\perp (1-x)^j (qx+1-x)^{n-j} - (x+1-x)^n \\
 &= \frac{M}{q^n} \sum_{j=0}^n A_j^\perp (1-x)^j \sum_{i=0}^{n-j} \binom{n-j}{i} q^i x^i (1-x)^{n-j-i} \\
 &\quad - \sum_{i=0}^n \binom{n}{i} x^i (1-x)^{n-i} \\
 &= \sum_{i=0}^n x^i (1-x)^{n-i} \left\{ \frac{M}{q^n} q^i \sum_{j=0}^{n-i} \binom{n-j}{i} A_j^\perp - \binom{n}{i} \right\}.
 \end{aligned}$$

Hence we get the following result.

Theorem 1.9. *Let C be an $(n, M; q)$ code. Then, for $1 \leq i \leq n$,*

$$\begin{aligned}
 A_i^\diamond(C) &= Mq^{i-n} \sum_{j=0}^{n-i} \binom{n-j}{i} A_j^\perp - \binom{n}{i} \\
 &= \binom{n}{i} (Mq^{i-n} - 1) + Mq^{i-n} \sum_{j=d^\perp}^{n-i} \binom{n-j}{i} A_j^\perp.
 \end{aligned}$$

From the definition and Theorem 1.9 we get the following corollary.

Corollary 1.2. *Let C be an $(n, M; q)$ code with minimum distance d and dual distance d^\perp . Then*

$$A_i^\diamond(C) = 0 \text{ for } 1 \leq i \leq d-1,$$

$$A_i^\diamond(C) \geq \max\left\{0, \binom{n}{i} (Mq^{i-n} - 1)\right\} \text{ for } d \leq i \leq n - d^\perp,$$

and

$$A_i^\diamond(C) = \binom{n}{i} (Mq^{i-n} - 1) \text{ for } n - d^\perp < i \leq n.$$

There is an alternative expression for $A_i^\diamond(C)$ which is more complicated, but quite useful.

For each set $E \subset \{1, 2, \dots, n\}$, define an equivalence relation \sim_E on C by $\mathbf{x} \sim_E \mathbf{y}$ if and only if $\chi(\mathbf{x}, \mathbf{y}) \subseteq E$ (that is, $x_i = y_i$ for all $i \notin E$). Let the set of equivalence classes be denoted X_E . If two vectors differ in at least one position outside E , then they are not equivalent. Therefore, the number of equivalence classes, that is, the size of X_E , is $q^{n-\#E}$.

Theorem 1.10. *Let C be an $(n, M; q)$ code. Then, for $1 \leq i \leq n$,*

$$A_j^\circ(C) = \frac{1}{M} \sum_{\substack{E \subset \{1, 2, \dots, n\} \\ \#E=j}} \sum_{U \in X_E} \#U(\#U - 1).$$

Proof. We count the number of elements in the set

$$V = \{(E, \mathbf{x}, \mathbf{y}) \mid E \subset \{1, 2, \dots, n\}, \#E = j, \mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}, \mathbf{x} \sim_E \mathbf{y}\}$$

in two ways. On one hand, for given E and an equivalence class $U \in X_E$, the pair (\mathbf{x}, \mathbf{y}) can be chosen in $\#U(\#U - 1)$ different ways. Hence, the the number of elements of V is given by

$$\#V = \sum_{\substack{E \subset \{1, 2, \dots, n\} \\ \#E=j}} \sum_{U \in X_E} \#U(\#U - 1). \tag{1.5}$$

On the other hand, for a given pair (\mathbf{x}, \mathbf{y}) of code words at distance $i \leq j$, E must contain the i elements in the support $\chi(\mathbf{x}, \mathbf{y})$ and $j - i$ of the $n - i$ elements outside the support. Hence, E can be chosen in $\binom{n-i}{j-i}$ ways. Since a pair (\mathbf{x}, \mathbf{y}) of code words at distance i can be chosen in $MA_i(C)$ ways, we get

$$\#V = \sum_{i=1}^j MA_i(C) \binom{n-i}{j-i} = MA_j^\circ(C). \tag{1.6}$$

Theorem 1.10 follows by combining (1.5) and (1.6). □

From Theorem 1.10 we can derive a lower bound on $A_j^\circ(C)$ which is sharper than (or sometimes equal to) the bound in Corollary 1.2.

First we need a simple lemma.

Lemma 1.4. *Let m_1, m_2, \dots, m_N be non-negative integers with sum M . Then*

$$\sum_{i=1}^N m_i^2 \geq \left(2 \left\lfloor \frac{M}{N} \right\rfloor + 1\right) \left(M - N \left\lfloor \frac{M}{N} \right\rfloor\right) + N \left\lfloor \frac{M}{N} \right\rfloor^2 \tag{1.7}$$

$$= M + \left(\left\lceil \frac{M}{N} \right\rceil - 1\right) \left(2M - N \left\lceil \frac{M}{N} \right\rceil\right), \tag{1.8}$$

with equality if and only if

$$\left\lfloor \frac{M}{N} \right\rfloor \leq m_i \leq \left\lceil \frac{M}{N} \right\rceil$$

for all i .

Proof. Let x_1, x_2, \dots, x_N be non-negative integers for which $\sum_{i=1}^N x_i^2$ is minimal. Without loss of generality, we may assume that $x_1 \leq x_i \leq x_N$ for all i . Suppose $x_N \geq x_1 + 2$. Let $y_1 = x_1 + 1$, $y_N = x_N - 1$, $y_i = x_i$ otherwise. Then, by the minimality of $\sum x_i^2$,

$$0 \leq \sum_{i=1}^N y_i^2 - \sum_{i=1}^N x_i^2 = (x_1 + 1)^2 - x_1^2 + (x_N - 1)^2 - x_N^2 = 2(x_1 - x_N + 1),$$

contradicting the assumption $x_N \geq x_1 + 2$. Therefore, we must have

$$x_N = x_1 + 1 \text{ or } x_N = x_1.$$

Let $\alpha = \lfloor M/N \rfloor$ and $M = N\alpha + \beta$ where $0 \leq \beta < N$. Then β of the x_i must have value $\alpha + 1$ and the remaining $N - \beta$ have value α and so

$$\sum_{i=1}^N x_i^2 = \beta(\alpha + 1)^2 + (N - \beta)\alpha^2 = (2\alpha + 1)\beta + N\alpha^2.$$

This proves (1.7). We have

$$\left\lceil \frac{M}{N} \right\rceil = \alpha \text{ if } \beta = 0, \text{ and } \left\lceil \frac{M}{N} \right\rceil = \alpha + 1 \text{ if } \beta > 0.$$

Hence (1.8) follows by rewriting (1.7). \square

Using Lemma 1.4, with the lower bound in the version (1.8), we see that the inner sum $\sum_{U \in X_E} \#U(\#U - 1)$ in Theorem 1.7 is lower bounded by

$$\sum_{U \in X_E} \#U(\#U - 1) \geq \left(\left\lceil \frac{M}{q^{n-j}} \right\rceil - 1 \right) \left(2M - q^{n-j} \left\lceil \frac{M}{q^{n-j}} \right\rceil \right),$$

independent of E . For E there are $\binom{n}{j}$ possible choices. Hence, we get the following bound.

Theorem 1.11. *Let C be an $(n, M; q)$ code. Then, for $1 \leq j \leq n$,*

$$A_j^\diamond(C) \geq \binom{n}{j} \left(\left\lceil \frac{M}{q^{n-j}} \right\rceil - 1 \right) \left(2 - \frac{q^{n-j}}{M} \left\lceil \frac{M}{q^{n-j}} \right\rceil \right).$$

1.3.4 Distance distribution of complementary codes

There is a close connection between the distance distributions of a code and its (set) complement. More general, there is a connection between the distance distributions of two disjoint codes whose union is a distance invariant code.

An $(n, M; q)$ code is called *distance invariant* if

$$\sum_{\mathbf{y} \in C} z^{d_H(\mathbf{x}, \mathbf{y})} = A_C(z)$$

for all $\mathbf{x} \in C$. In particular, any linear code is distance invariant. However, a code may be distance invariant without being linear.

Example 1.1. A simple example of a non-linear distance invariant code is the code

$$\{(1000), (0100), (0010), (0001)\}.$$

Theorem 1.12. Let the $(n, M_1; q)$ code C_1 and the $(n, M_2; q)$ code C_2 be disjoint codes such that $C_1 \cup C_2$ is distance invariant. Then,

$$M_1 \{A_{C_1 \cup C_2}(z) - A_{C_1}(z)\} = M_2 \{A_{C_1 \cup C_2}(z) - A_{C_2}(z)\}.$$

Proof. Since $C_1 \cup C_2$ is distance invariant, we have

$$\begin{aligned} M_1 A_{C_1 \cup C_2}(z) &= \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_1 \cup C_2} z^{d_H(\mathbf{x}, \mathbf{y})} \\ &= \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_1} z^{d_H(\mathbf{x}, \mathbf{y})} + \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_2} z^{d_H(\mathbf{x}, \mathbf{y})} \\ &= M_1 A_{C_1}(z) + \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_2} z^{d_H(\mathbf{x}, \mathbf{y})}, \end{aligned}$$

and so

$$M_1 \{A_{C_1 \cup C_2}(z) - A_{C_1}(z)\} = \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_2} z^{d_H(\mathbf{x}, \mathbf{y})}.$$

Similarly,

$$M_2 \{A_{C_1 \cup C_2}(z) - A_{C_2}(z)\} = \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_2} z^{d_H(\mathbf{x}, \mathbf{y})},$$

and the theorem follows. \square

If $C_2 = \overline{C_1}$, then the conditions of Theorem 1.12 are satisfied. Since $C_1 \cup C_2 = F_q^n$ we have $M_2 = q^n - M_1$ and $A_{C_1 \cup C_2}(z) = (1 + (q-1)z)^n$. Hence we get the following corollary.

Corollary 1.3. Let C be an $(n, M; q)$ code. Then

$$A_{\overline{C}}(z) = \frac{M}{q^n - M} A_C(z) + \frac{q^n - 2M}{q^n - M} (1 + (q-1)z)^n.$$

From Corollary 1.3 we immediately get the following corollary.

Corollary 1.4. *Let C be an $(n, M; q)$ code. Then, for $0 \leq i \leq n$, we have*

$$A_i(\overline{C}) = \frac{M}{q^n - M} A_i(C) + \frac{q^n - 2M}{q^n - M} \binom{n}{i} (q-1)^i.$$

Using Corollary 1.4 we get the following.

Corollary 1.5. *Let C be an $(n, M; q)$ code. Then, for $1 \leq i \leq n$, we have*

$$A_i^\diamond(\overline{C}) = \frac{M}{q^n - M} A_i^\diamond(C) + \frac{q^n - 2M}{q^n - M} \binom{n}{i} (q^i - 1).$$

Proof. We have

$$\begin{aligned} A_i^\diamond(\overline{C}) &= \sum_{j=1}^i A_i(\overline{C}) \binom{n-j}{n-i} \\ &= \frac{M}{q^n - M} \sum_{j=1}^i A_j(C) \binom{n-j}{n-i} + \frac{q^n - 2M}{q^n - M} \sum_{j=1}^i \binom{n-j}{n-i} \binom{n}{j} (q-1)^j \\ &= \frac{M}{q^n - M} A_i^\diamond(C) + \frac{q^n - 2M}{q^n - M} \sum_{j=1}^i \binom{n}{i} \binom{i}{j} (q-1)^j \\ &= \frac{M}{q^n - M} A_i^\diamond(C) + \frac{q^n - 2M}{q^n - M} \binom{n}{i} (q^i - 1). \end{aligned} \quad \square$$

1.4 Weight distribution of linear codes

1.4.1 Weight distribution

Let

$$A_i^w = A_i^w(C) = \#\{\mathbf{x} \in C \mid w_H(\mathbf{x}) = i\}.$$

The sequence $A_0^w, A_1^w, \dots, A_n^w$ is known as the *weight distribution* of C and

$$A_C^w(z) = \sum_{i=0}^n A_i^w z^i$$

is the *weight distribution function* of C .

We note that $d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} - \mathbf{y})$. If C is linear, then $\mathbf{x} - \mathbf{y} \in C$ when $\mathbf{x}, \mathbf{y} \in C$. Hence we get the following useful result.

Theorem 1.13. *For a linear code C we have $A_i(C) = A_i^w(C)$ for all i and $A_C(z) = A_C^w(z)$.*

If C and C' are equivalent codes, then clearly $A_i(C) = A_i(C')$. In particular, for the study of the weight distribution of linear codes we may therefore without loss of generality assume that the code is systematic if we so wish.

1.4.2 Weight distribution of *-extended codes

The *-operation for linear codes was defined on page 8. The code S_k is a constant weight code, that is, all non-zero code words have the same weight, namely q^{k-1} .

Therefore, $A_{C^*}(z)$ only depends on $A_C(z)$. In fact

$$A_{C^*}(z) - 1 = z^{q^{k-1}}(A_C(z) - 1)$$

since each non-zero vector is extended by a part of weight q^{k-1} .

1.4.3 MacWilliams's theorem

The following theorem is known as *MacWilliams's theorem*.

Theorem 1.14. *Let C be a linear $[n, k; q]$ code. Then*

$$A_i^\perp(C) = A_i(C^\perp).$$

Equivalently,

$$A_{C^\perp}(z) = \frac{1}{q^k}(1 + (q - 1)z)^n A_C\left(\frac{1 - z}{1 + (q - 1)z}\right).$$

Proof. We prove this for q a prime, using Theorem 1.5. The proof for general prime power q is similar, using Theorem 1.6. First we show that

$$\sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} = \begin{cases} M & \text{if } \mathbf{u} \in C^\perp, \\ 0 & \text{if } \mathbf{u} \notin C^\perp. \end{cases} \tag{1.9}$$

If $\mathbf{u} \in C^\perp$, then $\mathbf{u} \cdot \mathbf{c} = 0$ and $\zeta^{\mathbf{u} \cdot \mathbf{c}} = 1$ for all $\mathbf{c} \in C$, and the result follows. If $\mathbf{u} \notin C^\perp$, then there exists a code word $\mathbf{c}' \in C$ such that $\mathbf{u} \cdot \mathbf{c}' \neq 0$ and hence $\zeta^{\mathbf{u} \cdot \mathbf{c}'} \neq 1$. Because of the linearity, $\mathbf{c} + \mathbf{c}'$ runs through C when \mathbf{c} does. Hence

$$\sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} = \sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot (\mathbf{c} + \mathbf{c}')} = \zeta^{\mathbf{u} \cdot \mathbf{c}'} \sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}}.$$

Hence $\sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} = 0$. This proves (1.9). By Theorem 1.5,

$$A_i^\perp(C) = \frac{1}{M^2} \sum_{\substack{\mathbf{u} \in \mathbb{Z}_q^n \\ w_H(\mathbf{u})=i}} \left| \sum_{\mathbf{c} \in C} \zeta^{\mathbf{u} \cdot \mathbf{c}} \right|^2 = \frac{1}{M^2} \sum_{\substack{\mathbf{u} \in C^\perp \\ w_H(\mathbf{u})=i}} M^2 = A_i(C^\perp). \quad \square$$

Corollary 1.6. *Let C be a linear $[n, k; q]$ code. Then*

$$d^\perp(C) = d(C^\perp).$$

1.4.4 A generalized weight distribution

Many generalizations of the weight distribution have been studied. One that is particularly important for error detection is the following.

Let C be an $[n, k]$ code and m a divisor of n . Let $A_{i_1, i_2, \dots, i_m}(C)$ be the number of vectors $(\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_m) \in C$ such that each part $\mathbf{x}_j \in GF(q)^{n/m}$ and $w_H(\mathbf{x}_j) = i_j$ for $j = 1, 2, \dots, m$. Further, let

$$A_C(z_1, z_2, \dots, z_m) = \sum_{i_1, i_2, \dots, i_m} A_{i_1, i_2, \dots, i_m}(C) z_1^{i_1} z_2^{i_2} \dots z_m^{i_m}.$$

For $m = 1$ we get the usual weight distribution function. Theorem 1.14 generalizes as follows.

Theorem 1.15. *Let C be a linear $[n, k; q]$ code. Then*

$$A_C(z_1, z_2, \dots, z_m) = q^{k-n} \left\{ \prod_{j=1}^m (1 + (q-1)z_j) \right\}^{\frac{n}{m}} A_{C^\perp}(z'_1, z'_2, \dots, z'_m)$$

where

$$z'_j = \frac{1 - z_j}{1 + (q-1)z_j}.$$

1.4.5 Linear codes over larger fields

There is an alternative expression for the weight distribution function that is useful for some applications. Let G be a $k \times n$ generator matrix over $GF(q)$. Let $m_G : GF(q)^k \rightarrow \mathcal{N} = \{0, 1, 2, \dots\}$, the *column count function*, be defined such that G contains exactly $m(\mathbf{x}) = m_G(\mathbf{x})$ columns equal to \mathbf{x} for all $\mathbf{x} \in GF(q)^k$. We use the following further notations:

$$[a]_b = \prod_{i=0}^{b-1} (q^a - q^i),$$

$$s(U, m) = \sum_{\mathbf{x} \in U} m(\mathbf{x}) \text{ for all } U \subseteq GF(q)^k,$$

\mathcal{S}_{kl} is the set of l dimensional subspaces of $GF(q)^k$,

$$\sigma_{kl}(m, z) = \sum_{U \in \mathcal{S}_{kl}} z^{s(\bar{U}, m)}, \text{ where } \bar{U} = GF(q)^k \setminus U,$$

$$\hat{U} = \{\mathbf{y} \in GF(q^r)^k \mid \mathbf{y} \cdot \mathbf{x} = 0 \text{ for } \mathbf{x} \in GF(q)^k \text{ if and only if } \mathbf{x} \in U\},$$

$$C_r = \{\mathbf{y}G \mid \mathbf{y} \in GF(q^r)^k\}, \text{ the code generated by } G \text{ over } GF(q^r),$$

$$C = C_1.$$

Theorem 1.16. *For $r \geq 1$ we have*

$$A_{C_r}(z) = \sum_{l=0}^k [r]_{k-l} \sigma_{kl}(m, z).$$

Proof. First we note that if $\mathbf{y} \in \hat{U}$, then

$$w_H(\mathbf{y}G) = \sum_{\mathbf{x} \in GF(q)^k} m(\mathbf{x}) w_H(\mathbf{y} \cdot \mathbf{x}) = \sum_{\mathbf{x} \in \bar{U}} m(\mathbf{x}) = s(\bar{U}, m).$$

Hence

$$A_{C_r}(z) = \sum_{l=0}^k \sum_{U \in \mathcal{S}_{kl}} \sum_{\mathbf{y} \in \hat{U}} z^{w_H(\mathbf{y}G)} = \sum_{l=0}^k \sum_{U \in \mathcal{S}_{kl}} z^{s(\bar{U}, m)} \sum_{\mathbf{y} \in \hat{U}} 1.$$

Since

$$\sum_{\mathbf{y} \in \hat{U}} 1 = [r]_{k-l},$$

the theorem follows. □

For $r = 1$, we get the following alternative expression for the weight distribution of C .

Corollary 1.7. *We have*

$$A_C(z) = 1 + \sum_{U \in \mathcal{S}_{k, k-1}} z^{s(\bar{U}, m)}.$$

1.4.6 Weight distribution of cosets

Theorem 1.17. *Let C be an $[n, k; q]$ code and S a proper coset of C . Let D be the $[n, k + 1; q]$ code containing C and S . Then*

$$A_S^w(z) = \frac{1}{q-1} \{A_D(z) - A_C(z)\}.$$

Proof. For each non-zero $a \in GF(q)$, $aS = \{ax \mid x \in S\}$ is also a proper coset of C and $A_{aS}^w(z) = A_S^w(z)$. Further $D = C \cup \bigcup_{a \neq 0} aS$ (disjoint union) and so

$$A_D(z) = A_C(z) + (q-1)A_S^w(z), \quad (1.10)$$

and the theorem follows. \square

Using the MacWilliams identity we get the following alternative expression.

Corollary 1.8. *Let C be an $[n, k; q]$ code and S a proper coset of C . Let D be the $[n, k+1; q]$ code containing C and S . Then*

$$A_S^w(z) = \frac{(1 + (q-1)z)^n}{q^{n-k}(q-1)} \left\{ qA_{D^\perp} \left(\frac{1-z}{1+(q-1)z} \right) - A_{C^\perp} \left(\frac{1-z}{1+(q-1)z} \right) \right\}.$$

Theorem 1.18. *Let C be an $[n, k; q]$ code and S a proper coset of C . Then*

$$A_S^w(z) \geq z^{n-k} A_C(z) \quad (1.11)$$

for all $z \in [0, 1]$.

Proof. We may assume without loss of generality that the code C is systematic. There exists a $\mathbf{v} \in S$ such that $S = \mathbf{v} + C$ and such that $\mathbf{v} = (\mathbf{0}|\mathbf{b})$ where $\mathbf{b} \in GF(q)^{n-k}$.

Let $(\mathbf{x}|\mathbf{x}') \in C$ where $\mathbf{x} \in GF(q)^k$ and $\mathbf{x}' \in GF(q)^{n-k}$. Then

$$\begin{aligned} w_H((\mathbf{x}|\mathbf{x}') + (\mathbf{0}|\mathbf{b})) &= w_H(\mathbf{x}) + w_H(\mathbf{x}' + \mathbf{b}) \\ &\leq w_H(\mathbf{x}) + n - k \\ &\leq w_H((\mathbf{x}|\mathbf{x}')) + n - k \end{aligned}$$

and so

$$z^{w_H((\mathbf{x}|\mathbf{x}')+(\mathbf{0}|\mathbf{b}))} \geq z^{n-k} z^{w_H((\mathbf{x}|\mathbf{x}'))}.$$

Summing over all $(\mathbf{x}|\mathbf{x}') \in C$, the theorem follows. \square

Corollary 1.9. *Let C be an $[n, k; q]$ code and D an $[n, k+1; q]$ code containing C . Then*

$$A_D(z) \geq \left\{ 1 + (q-1)z^{n-k} \right\} A_C(z).$$

Proof. Let $S \subset D$ be a proper coset of C . By (1.10) and Theorem 1.18 we have

$$A_D(z) = A_C(z) + (q-1)A_S^w(z) \geq A_C(z) + (q-1)z^{n-k} A_C(z). \quad \square$$

Theorem 1.19. *Let C be an $[n, k; q]$ code and S a proper coset of C . Then*

$$A_S^w(z) \leq \frac{1 - y^{k+1}}{1 + (q - 1)y^{k+1}} A_C(z)$$

for all $z \in [0, 1]$, where $y = (1 - z)/(1 + (q - 1)z)$.

Theorem 1.20. *Let C be an $[n, k; q]$ code and D an $[n, k + 1; q]$ code which contains C . Then*

$$A_C(z) \geq \frac{1 + (q - 1)y^{k+1}}{q} A_D(z)$$

for all $z \in [0, 1]$, where $y = (1 - z)/(1 + (q - 1)z)$.

Proof. By Corollary 1.9 we get

$$\begin{aligned} A_C(z) &= q^{k-n} (1 + (q - 1)z)^n A_{C^\perp}(y) \\ &\geq q^{k-n} (1 + (q - 1)z)^n (1 + (q - 1)y^{k+1}) A_{D^\perp}(y) \\ &= q^{-1} (1 + (q - 1)y^{k+1}) A_D(z) \\ &= q^{-1} (1 + (q - 1)y^{k+1}) (A_C(z) + (q - 1)A_S^w(z)) \end{aligned}$$

and the theorems follow. □

Corollary 1.10. *If C is an $[n, k; q]$ code and $k < n$, then*

$$A_C(z) \geq \frac{(1 + (q - 1)z)^n}{q^{n-k}} \prod_{j=k+1}^n (1 + (q - 1)y^j),$$

for all $z \in [0, 1]$, where $y = (1 - z)/(1 + (q - 1)z)$.

Proof. The corollary follows from Theorem 1.20 by induction on k . □

1.4.7 Counting vectors in a sphere

The sphere $S_t(\mathbf{x})$ of radius t around a vector $\mathbf{x} \in GF(q)^n$ is the set of vectors within Hamming distance t of \mathbf{x} , that is

$$S_t(\mathbf{x}) = \{\mathbf{y} \in GF(q)^n \mid d_H(\mathbf{x}, \mathbf{y}) \leq t\}.$$

Let $N_t(i, j)$ be the number of vectors of weight j in a sphere of radius t around a vector of weight i .

Theorem 1.21. *We have*

$$N_t(i, j) = \sum_{e=|i-j|}^t \sum_{\delta=\max(i,j)-e}^{\min(\lfloor \frac{i+j-e}{2} \rfloor, n-e)} \binom{n-i}{\beta} \frac{i!}{\gamma! \delta! \epsilon!} (q-1)^\beta (q-2)^\epsilon$$

where $\beta = e - i + \delta$, $\gamma = e - j + \delta$, $\epsilon = i + j - e - 2\delta$.

Proof. Let $w_H(\mathbf{x}) = i$ and let $\mathbf{y} \in S_t(\mathbf{x})$ such that $w_H(\mathbf{y}) = j$. Let

$$\begin{aligned} \alpha &= \#\{l \mid x_l = y_l = 0\}, \\ \beta &= \#\{l \mid x_l = 0, y_l \neq 0\}, \\ \gamma &= \#\{l \mid x_l \neq 0, y_l = 0\}, \\ \delta &= \#\{l \mid x_l = y_l \neq 0\}, \\ \epsilon &= \#\{l \mid x_l \neq 0, y_l \neq 0, x_l \neq y_l\}. \end{aligned} \tag{1.12}$$

Then

$$\begin{aligned} i &= w_H(\mathbf{x}) = \gamma + \delta + \epsilon, \\ j &= w_H(\mathbf{y}) = \beta + \delta + \epsilon, \\ e &= d_H(\mathbf{x}, \mathbf{y}) = \beta + \gamma + \epsilon, \\ n &= \alpha + \beta + \gamma + \delta + \epsilon. \end{aligned} \tag{1.13}$$

Hence

$$\begin{aligned} \beta &= e - i + \delta, \\ \gamma &= e - j + \delta, \\ \epsilon &= i + j - e - 2\delta. \end{aligned} \tag{1.14}$$

Further,

$$\begin{aligned} |i - j| &\leq e \leq t, \\ \delta &= i - e + \beta \geq i - e, \\ \delta &= j - e + \gamma \geq j - e, \\ \delta &= n - e - \alpha \leq n - e, \\ 2\delta &= i + j - e - \epsilon \leq i + j - e. \end{aligned} \tag{1.15}$$

On the other hand, if e and δ are integers such that (1.15) is satisfied, then there are

$$\binom{n-i}{\beta} \frac{i!}{\gamma! \delta! \epsilon!} (q-1)^\beta (q-2)^\epsilon$$

ways to choose \mathbf{y} such that (1.12)–(1.14) are satisfied. □

For $q = 2$, the terms in the sum for $N_t(i, j)$ are 0 unless $\epsilon = 0$. We get the following simpler expression in this case:

$$N_t(i, j) = \sum_{\gamma=\max(0, i-j)}^{\lfloor \frac{i+j}{2} \rfloor} \binom{n-i}{\gamma+j-i} \binom{i}{\gamma}. \tag{1.16}$$

1.4.8 Bounds on the number of code words of a given weight

Some useful upper bounds on A_i for a linear code are given by the next theorem.

Theorem 1.22. *Let C be a linear $[n, k, d = 2t + 1; q]$ code. If $N_t(i, j) > 0$, then*

$$A_i \leq \frac{\binom{n}{j}}{N_t(i, j)}(q-1)^j.$$

In particular, for $d \leq i \leq \lfloor \frac{n}{2} \rfloor$ we have

$$A_i \leq \frac{\binom{n}{i}}{\binom{n-i+t}{t}}(q-1)^{i-t} \leq \frac{\binom{n}{i}}{\binom{\lfloor \frac{n}{2} \rfloor + t}{t}}(q-1)^{i-t},$$

and, for $\lceil \frac{n}{2} \rceil \leq i \leq n-t$,

$$A_i \leq \frac{\binom{n}{i}}{\binom{i+t}{t}}(q-1)^i \leq \frac{\binom{n}{i}}{\binom{\lceil \frac{n}{2} \rceil + t}{t}}(q-1)^i.$$

Proof. Counting all vectors of weight j and Hamming distance at most t from a code word of weight i we get

$$A_i N_t(i, j) \leq \binom{n}{j}(q-1)^j.$$

In particular, $N_t(i, i-t) = \binom{i}{t} > 0$ for all $i \geq d$, and so

$$A_i \leq \frac{\binom{n}{i-t}}{\binom{i}{t}}(q-1)^{i-t} = \frac{\binom{n}{i}}{\binom{n-i+t}{t}}(q-1)^{i-t}.$$

Similarly, $N_t(i, i+t) = \binom{n-i}{t}(q-1)^t > 0$ for $d \leq i \leq n-t$ and so

$$A_i \leq \frac{\binom{n}{i+t}(q-1)^{i+t}}{\binom{n-i}{t}(q-1)^t} = \frac{\binom{n}{i}}{\binom{i+t}{t}}(q-1)^i. \quad \square$$

Theorem 1.23. *For an $[n, k; q]$ code C we have $A_n \leq (q-1)^k$.*

Proof. Since equivalent codes have the same weight distribution, we may assume without loss of generality that the code is systematic, that is, it is generated by a matrix

$$G = (I_k | P) = \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_k \end{pmatrix}$$

where I_k is the $k \times k$ identity matrix, P is a $k \times (n-k)$ matrix, and $\mathbf{g}_1, \dots, \mathbf{g}_k$ are the rows of G . If $\mathbf{c} = \sum_{i=1}^k a_i \mathbf{g}_i$ has weight n , then in particular $a_i = c_i \neq 0$ for $1 \leq i \leq k$. Hence there are at most $(q-1)^k$ such \mathbf{c} . \square

There are many codes for which we have $A_n = (q - 1)^k$. For example, this is the case for any code that has a generator matrix where all the columns have weight one.

1.5 The weight hierarchy

For a linear $[n, k; q]$ code C and any r , where $1 \leq r \leq k$, the r -th minimum support weight is defined by

$$d_r = d_r(C) = \min \left\{ \#\chi(D) \mid D \text{ is an } [n, r; q] \text{ subcode of } C \right\}.$$

In particular, the minimum distance of C is d_1 . The *weight hierarchy* of C is the set $\{d_1, d_2, \dots, d_k\}$. The weight hierarchy satisfies the following inequality:

$$d_r \geq d_{r-1} \left(1 + \frac{q-1}{q^r - q} \right). \quad (1.17)$$

In particular, we have

$$d_r \geq d_{r-1} + 1. \quad (1.18)$$

An upper bound that follows from (1.18) is the *generalized Singleton bound*

$$d_r \leq n - k + r. \quad (1.19)$$

1.6 Principles of error detection

1.6.1 Pure detection

Consider what happens when a code word \mathbf{x} from an (n, M) code C is transmitted over a channel K and errors occur during transmission. If the received vector \mathbf{y} is not a code word we immediately realize that something has gone wrong during transmission, we *detect* that errors have occurred. However, it may happen that the combination of errors is such that the received vector \mathbf{y} is also a code word. In this case we have no way to tell that the received code word is not the sent code word. Therefore, we have an *undetected error*. We let $P_{ue} = P_{ue}(C, K)$ denote the probability that this happens. It is called the *probability of undetected error*. If $P(\mathbf{x})$ is the probability that \mathbf{x} was sent and $P(\mathbf{y}|\mathbf{x})$ is the probability that \mathbf{y} is received, given that \mathbf{x} was sent, then

$$P_{ue}(C, K) = \sum_{\mathbf{x} \in C} P(\mathbf{x}) \sum_{\mathbf{y} \in C \setminus \{\mathbf{x}\}} P(\mathbf{y}|\mathbf{x}).$$

In most cases we will assume that each code word is equally likely to be sent, that is, $P(\mathbf{x}) = \frac{1}{M}$. Under this assumption we get

$$P_{\text{ue}}(C, K) = \frac{1}{M} \sum_{\mathbf{x} \in C} \sum_{\mathbf{y} \in C \setminus \{\mathbf{x}\}} P(\mathbf{y}|\mathbf{x}).$$

The quantity $P_{\text{ue}}(C, K)$ is a main parameter for describing how well C performs on the channel K , and it is the main subject of study in this book. In Chapter 2, we study $P_{\text{ue}}(C, K)$ for the q -ary symmetric channel, in Chapter 3 we describe results that are particular for the binary symmetric channel, in Chapter 4 we study other channels.

Remark 1.1. It is easy to show that for any channel K with additive noise and any coset S of a linear code C we have $P_{\text{ue}}(C, K) = P_{\text{ue}}(S, K)$.

1.6.2 Combined correction and detection

In some applications we prefer to use some of the power of a code to correct errors and the remaining power to detect errors. Suppose that C is an $(n, M; q)$ code capable of correcting all error patterns with t_0 or less errors that can occur on the channel and suppose that we use the code to correct all error patterns with t errors or less, where $t \leq t_0$. Let $M_t(\mathbf{x})$ be the set of all vectors \mathbf{y} such that $d_{\text{H}}(\mathbf{x}, \mathbf{y}) \leq t$ and such that \mathbf{y} can be received when \mathbf{x} is sent over the channel. For two distinct $\mathbf{x}_1, \mathbf{x}_2 \in C$, the sets $M_t(\mathbf{x}_1), M_t(\mathbf{x}_2)$ are disjoint. If $\mathbf{y} \in M_t(\mathbf{x})$ is received, we decode into \mathbf{x} . If $\mathbf{y} \notin M_t(\mathbf{x})$ for all $\mathbf{x} \in C$, then we detect an error.

Suppose that \mathbf{x} is sent and \mathbf{y} is received. There are then three possibilities:

- (1) $\mathbf{y} \in M_t(\mathbf{x})$. We then decode, correctly, into \mathbf{x} .
- (2) $\mathbf{y} \notin M_t(\mathbf{x}')$ for all $\mathbf{x}' \in C$. We then detect an error.
- (3) $\mathbf{y} \in M_t(\mathbf{x}')$ for some $\mathbf{x}' \in C \setminus \{\mathbf{x}\}$. We then decode erroneously into \mathbf{x}' , and we have an undetectable error.

Let $P_{\text{ue}}^{(t)} = P_{\text{ue}}^{(t)}(C, K)$ denote the probability that we have an undetectable error. As above we get

$$P_{\text{ue}}^{(t)}(C, K) = \sum_{\mathbf{x} \in C} P(\mathbf{x}) \sum_{\mathbf{x}' \in C \setminus \{\mathbf{x}\}} \sum_{\mathbf{y} \in M_t(\mathbf{x}')} P(\mathbf{y}|\mathbf{x}).$$

Assuming that $P(\mathbf{x}) = \frac{1}{M}$ for all $\mathbf{x}' \in C$, we get

$$P_{\text{ue}}^{(t)}(C, K) = \frac{1}{M} \sum_{\mathbf{x} \in C} \sum_{\mathbf{x}' \in C \setminus \{\mathbf{x}\}} \sum_{\mathbf{y} \in M_t(\mathbf{x}')} P(\mathbf{y}|\mathbf{x}).$$

1.7 Comments and references

1.1 Most of this material can be found in most text books on error-correcting codes, see the general bibliography. However, many of the books restrict themselves to binary codes.

1.2 Again, this is mainly standard material.

1.3 Some of this material is standard. Most textbooks restrict their presentation to linear codes and, therefore, to the weight distribution.

Theorem 1.3 is due to Pless (1963).

Theorems 1.5 and 1.6 are due to Delsarte (1972).

Binomial moments seems to have been used for the first time by MacWilliams (1963). Possibly the first application to error detection is by Kløve (1984d). A survey on binomial moments was given by Dodunekova (2003b).

Theorem 1.9 and Corollary 1.2 were given in Kløve and Korzhik (1995, pp. 51–52) in the binary case. For general q , they were given by Dodunekova (2003b).

Theorems 1.10 and 1.11 is due to AbdelGhaffar (1997).

Theorem 1.12 is essentially due to AbdelGhaffar (2004). Corollary 1.3 (for $q = 2$) was first given by Fu, Kløve, and Wei (2003), with a different proof.

1.4 Theorem 1.14 is due to MacWilliams (1963). Theorem 1.15 (for $q = 2$) was given by Kasami, Fujiwara, and Lin (1986).

Theorem 1.16 is from Kløve (1992).

Theorem 1.17 and Corollary 1.8 are due to Assmus and Mattson (1978).

Theorem 1.18 is essentially due to Ancheti (1981).

Theorem 1.19 with $q = 2$ is due to Sullivan (1967). An alternative proof and generalization to general q was given by Redinbo (1973). Further results are given in Kløve (1993), Kløve (1994b), Kløve (1996c).

We remark that the weight distribution of cosets can be useful in the wire-tap channel area, see Wyner (1975) and Korzhik and Yakovlev (1992).

Theorem 1.21 is essentially due to MacWilliams (1963). In the present form it was given in Kløve (1984a).

Theorem 1.22 was given in Kløve and Korzhik (1995, Section 2.2). Special cases were given implicitly in Korzhik and Fink (1975) and Kasami, Kløve, and Lin (1983).

Theorem 1.23 is due to Kløve (1996a).

The weight hierarchy (under a different name) was first studied by

Helleseth, Kløve, and Mykkeltveit (1977). The r -th minimum support weight is also known as r -th *generalized Hamming weight*, see Wei (1991). The inequality (1.17) was shown by Helleseth, Kløve, and Ytrehus (1992) (for $q = 2$) and Helleseth, Kløve, and Ytrehus (1993) (for general q).

- 1.6. A more detailed discussion of combined error detection and correction is found for example in Kløve (1984a).