1.1 Basic concepts

> Although coding theory has its origin in an engineering problem, the subject has developed by using more and more sophisticated mathematical techniques.
>
> F. J. MacWilliams and N. J. A. Sloane, [11], p. vi.

Essential points

- The definition of block code and some basic related notions (codewords, dimension, transmission rate, minimum distance, equivalence criteria for block codes).
- The definition of decoder and of the correcting capacity of a decoder.
- The minimum distance decoder and its error-reduction factor.
- The archtypal Hamming code [7,4,3] and its computational treatment.
- Basic dimension upper bounds (Singleton and Hamming) and the notions of MDS codes and perfect codes.
- The dimension lower bound of Gilbert.

Introductory remarks

> The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.
>
> C. E. Shannon 1948, [21].

Messages generated by an *information source* often can be modelled as a *stream* $s_1, s_2, ...$ of *symbols* chosen from a finite set S called the *source alphabet*. Moreover, usually we may assume that the time t_s taken by the source to generate a symbol is the same for all symbols.

To send messages we need a *communications channel*. A communications channel can often be modelled as a device that takes a stream of symbols chosen from a finite set T, that we will call the *channel* (or *transmission*) *alphabet*, and pipes them to its destination (called the *receiving end* of the

channel) in some physical form that need not concern us here.[1] As a result, a stream of symbols chosen from T arrives at the receiving end of the channel.

The channel is said to be *noiseless* if the sent and received symbols always agree. Otherwise it is said to be *noisy*, as real channels almost always are due to a variety of physical phenomena that tend to distort the physical representation of the symbols along the channel.

The transfer from the source to the *sending end* of the channel requires an *encoder*, that is, a function $f : S \to T^*$ from the set of source symbols into the set T^* of finite sequences of transmission symbols. Since we do not want to loose information at this stage, we will always assume that encoders are injective. The elements of the image of f are called the *code-words* of the encoder.

In this text we will consider only *block encoders*, that is, encoders with the property that there exists a positive integer n such that $C \subseteq T^n$, where $C = f(S)$ is the set of code words of the encoder. The integer n is called the *length* of the block encoder.

For a block encoding scheme to make sense it is necessary that the source generates the symbols at a rate that leaves enough time for the opperation of the encoder and the channel transmission of the code-words. We always will assume that this condition is satisfied, for this requirement is taken into account in the design of communications systems.

Since for a block encoder the map $f : S \to C$ is bijective, we may consider as equivalent the knowledge of a source symbol and the corresponding code word. Generally speaking this equivalence holds also at the algorithmic level, since the computation of f or of its inverse usually can be efficiently managed. As a consequence, it will be possible to phrase the main issues concerning block encoders in terms of the set C and the properties of the channel. In the beginning in next section, we adopt this point of view as a general starting point of our study of block codes and, in particular, of the decoding notions and problems.

1.1 Remarks. The set $\{0, 1\}$ is called the *binary alphabet* and it is very widely used in communications systems. Its two symbols are called *bits*. With the addition and multiplication modulo 2, it coincides with the field $\mathbb{Z}_2$ of *binary digits*. The transposition of the two bits ($0 \mapsto 1, 1 \mapsto 0$) is called *negation*. Note that the negation of a bit b coincides with $1 + b$.

1.2 *Example*. Consider the $\mathsf{Rep}(3)$ encoder f considered in the Introduction (page 2). In this case S and T are the binary alphabet and $C = \{000, 111\}$, respectively. Note that the inverse of the bijection $f : S \to C$ is the map $000 \mapsto 0, 111 \mapsto 1$, which can be defined as $x \mapsto x_1$.

[1] The interested reader may consult [19, 5].

Block codes

> Only *block* codes for correcting *random* errors are discussed
> F. J. MacWilliams and N. J. A. Sloane, [11], p. vii.

Let $T = \{t_1, \ldots, t_q\}$ ($q \geqslant 2$) be the channel alphabet. By a (block) *code* of length n we understand any non-empty subset $C \subseteq T^n$. If we want to refer to q explicitely, we will say that C is q-*ary* (*binary* if $q = 2$, *ternary* if $q = 3$). The elements of C will be called *vectors*, *code-words* or simply *words*.

Usually the elements of T^n are written in the form $x = (x_1, \ldots, x_n)$, where $x_i \in T$. Since the elements of T^n can be seen as length n sequences of elements of T, the element x will also be written as $x_1 x_2 \ldots x_n$ (concatenated symbols), especially when $T = \{0, 1\}$.

If $x \in T^n$ and $x' \in T^{n'}$, the expression $x|x'$ will be used as an alternative notation for the element $(x, x') \in T^{n+n'}$. Similar notations such as $x|x'|x''$ will be used without further explanation.

Dimension, transmission rate and channel coding

If C is a code of length n, we will set $k_C = \log_q(|C|)$ and we will say that k_C is the *dimension* of C. The quotient $R_C = k_C/n$ is called *transmission rate*, or simply *rate*, of C.

1.3 Remarks. These notions can be clarified in terms of the basic notions explained in the first subsection (Introductory remarks). Indeed, $\log_q(|T^n|) = \log_q(q^n) = n$ is the number of transmission symbols of any element of T^n. So $k_C = \log_q(|C|) = \log_q(|S|)$ can be seen as the number of transmission symbols that are needed to capture the information of a source symbol. Consequently, to send the information content of k_C transmission symbols (which, as noted, amounts to a source symbol) we need n transmission symbols, and so R_C represents the proportion of source information contained in a code word before transmission. An interesting and useful special case occurs when $S = T^k$, for some positive integer k. In this case the source symbols are already resolved explicitely into k transmission symbols and we have $k_C = k$ and $R_C = k/n$. □

If C has length n and dimension k (respectively $|C| = M$), we say that C has type $[n, k]$ (respectively type (n, M)). If we want to have q explicitely in the notations, we will write $[n, k]_q$ or $(n, M)_q$. We will often write $C \sim [n, k]$ to denote that the type of C is $[n, k]$, and similar notations will be used for the other cases.

Minimum distance

> Virtually all research on error-correcting codes has been based on the Hamming metric.
>
> W.W. Peterson and E.J. Weldon, Jr., [16], p. 307.

Given $x, y \in T^n$, the *Hamming distance* between x and y, which we will denote $\mathsf{hd}(x, y)$, is defined by the formula

$$\mathsf{hd}(x, y) = |\{i \in 1..n \mid x_i \neq y_i\}|.$$

In other words, it is the number of positions i in which x and y differ.

E.1.1. Check that hd is a distance on T^n. Recall that $d : X \times X \to \mathbb{R}$ is said to be a distance on the set X if, for all $x, y, z \in X$,

1) $d(x, y) \geqslant 0$, with equality if and only if $x = y$;
2) $d(y, x) = d(x, y)$; and
3) $d(x, y) \leqslant d(x, z) + d(z, y)$.

The last relation is called *triangle inequality*.

E.1.2. Let $x, y \in \mathbb{Z}_2^n$ be two binary vectors. Show that

$$\mathsf{hd}(x, y) = |x| + |y| - 2|x \cdot y|$$

where $|x|$ is the number of non-zero entries of x (it is called the *weight* of x) and $x \cdot y = (x_1 y_1, \cdots, x_n y_n)$. □

We will set $d = d_C$ to denote the minimum of the distances $\mathsf{hd}(c, c')$, where $c, c' \in C$ and $c \neq c'$, and we will say that d is the *minimum distance* of C. Note that for the minimum distance to be defined it is necessary that $|C| \geqslant 2$. For the codes C that have a single word, we will see that it is convenient to put $d_C = n + 1$ (see the Remark 1.12).

We will say that a code C is of type $[n, k, d]$ (or of type (n, M, d)), if C has length n, minimum distance d, and its dimension is k (respectively $|C| = M$). If we want to have q explicitely in the notations, we will write $[n, k, d]_q$ or $(n, M, d)_q$. In the case $q = 2$ it is usually omitted. Sometimes we will write $C \sim [n, k, d]_q$ to denote that the type of C is $[n, k, d]_q$, and similar notations will be used for the other cases.

The rational number $\delta_C = d_C/n$ is called *relative distance* of C.

E.1.3. Let C be a code of type (n, M, d). Check that if $k = n$, then $d = 1$. Show also that if $M > 1$ (in order that d is defined) and $d = n$ then $M \leqslant q$, hence $k_C \leqslant 1$.

1.4 *Example* (A binary code (8,20,3))**.** Let C be the binary code $(8, 20)$ consisting of 00000000, 11111111 and all cyclic permutations of 10101010, 11010000 and 11100100. From the listing **Cyclic shifts of a vector** it is easy to infer that $d_C = 3$. Thus C is a code of type $(8, 20, 3)$.

Library | Cyclic shifts of a vector

```
cyclic_shift( x:Vector ) := [ x_length(x) ] | take(x,length(x)−1)
cyclic_shifts( x:Vector ) :=
begin
   local X={x}, y=cyclic_shift(x)
   while y≠x do
      X=X|{y}
      y=cyclic_shift(y)
   end
   X
end
```

```
\careta=[1,1,0,1,0,0,0,0]; X=cyclic_shifts(a);
b=[1,1,1,0,0,1,0,0]; Y=cyclic_shifts(b);
c=[1,0,1,0,1,0,1,0]; Z=cyclic_shifts(c);
{hd(a, x) with x in tail(X)} ⟶ {4, 4, 4, 6, 4, 4, 4}
{hd(a,y) with y in Y} ⟶ {3, 3, 5, 3, 5, 7, 3, 3}
{hd(a,z) with z in Z} ⟶ {5, 3}
{hd(b,y) with y in tail(Y)} ⟶ {4, 6, 4, 4, 4, 6, 4}
{hd(b,z) with z in Z} ⟶ {4, 4}
{hd(c,z) with z in tail(Z)} ⟶ {8}
```

1.5 Remark. A code with minimum distance d *detects* up to $d-1$ errors, in the sense that the introduction of a number of errors between 1 and $d-1$ in the transmission gives rise to a word that is not in C. Note that $d-1$ is the highest integer with this property (by definition of d).

> However, [the Hamming distance] is not the only possible and indeed may not always be the most appropriate. For example, in $(F_{10})^3$ we have $d(428, 438) = d(428, 468)$, whereas in practice, e.g. in dialling a telephone number, it might be more sensible to use a metric in which 428 is closer to 438 than it is to 468.
>
> R. Hill, [9], p. 5.

Equivalence criteria

We will say that two codes C and C' of length n are *strictly equivalent* if C' can be obtained by permuting the entries of all vectors in C with some fixed permutation. This relation is an equivalence relation on the set of all codes of length n and by definition we see that it is the equivalence relation

that corresponds to the natural action of $\mathcal{S}_n$ on T^n, and hence also on subsets of T^n.

In the discussion of equivalence it is convenient to include certain permutations of the alphabet T in some specified positions. This idea can be formalized as follows. Let $\Gamma = (\Gamma_1, \ldots, \Gamma_n)$, where Γ_i is a subgroup of permutations of T, that is, a subgroup of $\mathcal{S}_q$ ($1 \leqslant i \leqslant n$). Then we say that two codes C and C' are Γ-*equivalent* if C' can be obtained from C by a permutation $\sigma \in \mathcal{S}_n$ applied, as before, to the entries of all vectors of C, followed by permutations $\tau_i \in \Gamma_i$ of the symbols of each entry i, $i = 1, \ldots, n$. If $\Gamma_i = \mathcal{S}_q$ for all i, instead of Γ-equivalent we will say $\mathcal{S}_q$*-equivalent*, or simply *equivalent*. In the case in which T is a finite field $\mathbb{F}$ and $\Gamma_i = \mathbb{F} - \{0\}$, acting on $\mathbb{F}$ by multiplication, instead of Γ-equivalent we will also say $\mathbb{F}$*-equivalent*, or $\mathbb{F}^*$*-equivalent*, or *scalarly equivalent*.

Note that the identity and the transposition of $\mathbb{Z}_2 = \{0, 1\}$ can be represented as the operations $x \mapsto x+0$ and $x \mapsto x+1$, respectively, and therefore the action of a sequence $\tau_1, \ldots, \tau_n$ of permutations of $\mathbb{Z}_2$ is equivalent to the addition of the vector $\tau \in \mathbb{Z}_2^n$ that corresponds to those permutations.

In general it is a relatively easy task, given Γ, to obtain from a code C other codes that are Γ-equivalent to C, but it is much more complicated to decide whether two given codes are Γ-equivalent.

E.1.4. Check that two (strongly) equivalent codes have the same parameters n, k and d.

E.1.5. Show that given a code $C \subseteq T^n$ and a symbol $t \in T$, there exists a code equivalent to C that contains the constant word $\boldsymbol{t}_n$.

E.1.6. Can there be codes of type $(n, q, n)_q$ which are not equivalent to the q-ary repetition code? How many non-equivalent codes of type $(n, 2, d)_2$ there are?

Decoders

The essential ingredient in order to use a code $C \subseteq T^n$ at the receiving end of a channel to reduce the errors produced by the channel noise is a *decoding function*. In the most general terms, it is a map

$$g\colon D \to C, \text{ where } C \subseteq D \subseteq T^n$$

such that $g(x) = x$ for all $x \in C$. The elements of D, the domain of g, are said to be g*-decodable*. By hypothesis, all elements of C are decodable. In case $D = T^n$, we will say that g is a *full decoder* (or a *complete decoder*).

We envisage g working, again in quite abstract terms, as follows. Given $x \in C$, we imagine that it is sent through a communications channel. Let

$y \in T^n$ be the vector received at the other end of the channel. Since the channel may be noisy, y may be different from x, and in principle can be any vector of T^n. Thus there are two possibilities:

- if $y \in D$, we will take the vector $x' = g(y) \in C$ as the decoding of y;
- otherwise we will say that y is *non-decodable*, or that a *decoder error* has occurred.

Note that the condition $g(x) = x$ for all $x \in C$ says that when a code word is received, the decoder returns it unchanged. The meaning of this is that the decoder is assuming, when the received word is a code-word, that it was the transmitted code-word and that no error occurred.

If we transmit x, and y is decodable, it can happen that $x' \neq x$. In this case we say that an (undetectable) *code error* has occurred.

Correction capacity

We will say that the decoder g has *correcting capacity* t, where t is a positive integer, if for any $x \in C$, and any $y \in T^n$ such that $\mathsf{hd}(x, y) \leqslant t$, we have $y \in D$ and $g(y) = x$.

1.6 *Example*. Consider the code $\mathsf{Rep}(3)$ and its decoder g considered in the Introduction (page 2). In this case $T = \{0, 1\}$, $C = \{000, 111\}$ and $D = T^3$, so that it is a full decoder. It corrects one error and undetectable code errors are produced when 2 or 3 bit-errors occur in a single code-word.

1.7 Remark. In general, the problem of decoding a code C is to construct D and g by means of efficient algorithms and in such a way that the correcting capacity is as high as possible.

Minimum distance decoder

Let us introduce some notation first. Given $w \in T^n$ and a non-negative integer r, we set

$$B(w, r) = \{z \in T^n \mid \mathsf{hd}(w, z) \leqslant r\}.$$

The set $B(w, r)$ is called the *ball of center* w and *radius* r.

If $C = \{x^1, \ldots, x^M\}$, let $D_i = B(x^i, t)$, where $t = \lfloor (d-1)/2 \rfloor$, with d the minimum distance of C. It is clear that $C \cap D_i = \{x^i\}$ and that $D_i \cap D_j = \emptyset$ if $i \neq j$ (by definition of t and the triangular inequality of the Hamming distance). Therefore, if we set $D_C = \bigsqcup_i D_i$, there is a unique map $g\colon D_C \to C$ such that $g(y) = x^i$ for all $y \in D_i$. By construction, g is a decoder of C and it corrects t errors. It is called the *minimum distance decoder* of C.

E.1.7. Check the following statements:

1. $g(y)$ is the word $x' \in C$ such that $\mathsf{hd}(y, x') \leqslant t$, if such an x' exists, and otherwise y is non-decodable for g.
2. If y is decodable and $g(y) = x'$, then

$$\mathsf{hd}(x, y) > t \text{ for all } x \in C - \{x'\}.$$

1.8 Remarks. The usefulness of the minimum distance decoder arises from the fact that in most ordinary situations the transmissions $x \mapsto y$ that lead to a decoder error ($y \notin D$), or to undetectable errors ($y \in D$, but $\mathsf{hd}(y, x) > t$) will in general be less likely than the transmissions $x \mapsto y$ for which y is decodable and $g(y) = x$.

To be more precise, the minimum distance decoder maximizes the likelihood of correcting errors if all the transmission symbols have the same probability of being altered by the channel noise and if the $q - 1$ possible errors for a given symbol are equally likely. If these conditions are satisfied, the channel is said to be a (q-ary) *symmetric channel*. Unless otherwise declared, henceforth we will understand that 'channel' means 'symmetric channel'.

From the computational point of view, the minimum distance decoder, as defined above, is inefficient in general, even if d_C is known, for it has to calculate $\mathsf{hd}(y, x)$, for $x \in C$, until $\mathsf{hd}(y, x) \leqslant t$, so that the average number of distances that have to be calculated is of the order of $|C| = q^k$. Note also that this requires having generated the q^k elements of C.

But we also have to say that the progress in block coding theory in the last fifty years can, to a considerable extent, be seen as a series of milestones that signal conceptual and algorithmic improvements enabling to deal with the minimum distance decoder, for large classes of codes, in ever more efficient ways. In fact, the wish to collect a representative sample of these brilliant achievements has guided the selection of the decoders presented in subsequent sections of this book, starting with next example.

1.9 *Example* (The *Hamming code* [7,4,3])**.** Let $K = \mathbb{Z}_2$ (the field of binary digits). Consider the K-matrix

$$R = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Note that the columns of R are the binary vectors of length 3 whose weight is at least 2. Writing I_r to denote the identity matrix of order r, let $G = I_4|R^T$ and $H = R|I_3$ (concatenate I_4 and R^T, and also R and I_3, by rows). Note

that the columns of H are precisely the seven non-zero binary vectors of length 3.

Let $S = K^4$ (the source alphabet) and $T = K$ (the channel alphabet). Define the block encoding $f : K^4 \to K^7$ by $u \mapsto uG = u|uR^T$. The image of this function is $C = \langle G \rangle$, the K-linear subspace spanned by the rows of G, so that C is a $[7, 4]$ code. Since

$$GH^T = (I_4|R^T)\left(\frac{R^T}{I_3}\right) = R^T + R^T = 0,$$

because the arithmetic is mod 2, we see that the rows of G, and hence the elements of C, are in the kernel of H^T. In fact,

$$C = \{y \in K^7 \mid yH^T = 0\},$$

as the right-hand side contains C and both expressions are K-linear subspaces of dimension 4. From the fact that all columns of H are distinct, it is easy to conclude that $d_C = 3$. Thus C has type $[7, 4, 3]$. Since $|C| \cdot \mathit{vol}(7, 1) = 2^4(1 + 7) = 2^8 = |K^8|$, we see that $D_C = K^7$.

As a decoding function we take the map $g : K^7 \to C$ defined by the following recipe:

1. Let $s = yH^T$ (this length 3 binary vector is said to be the *syndrome* of the vector y).
2. If $s = 0$, return y (as we said above, $s = 0$ is equivalent to say that $y \in C$).
3. If $s \neq 0$, let j be the index of s as a row of H^T.
4. Negate the j-th bit of y.
5. Return the first four components of y.

Let us show that this decoder, which by construction is a full decoder ($D = K^7$), *coincides with the minimum distance decoder*. Indeed, assume that $x \in C$ is the vector that has been sent. If there are no errors, then $y = x$, $s = 0$, and the decoder returns x. Now assume that the j-th bit of $x \in C$ is changed during the transmission, and that the received vector is y. We can write $y = x + e_j$, where e_j is the vector with 1 on the j-th entry and 0 on the remaining ones. Then $s = yH^T = xH^T + e_jH^T = e_jH^T$, which clearly is the j-th row of H^T. Hence $g(y) = y + e_j = x$ (note that the operation $y \mapsto y + e_j$ is equivalent to negating the j-th bit of y). □

The expression of this example in **WIRIS/*cc*** is explained in the listing **Hamming code [7,4,3]**. The **WIRIS** elements that are used may be consulted in the Appendix, or in the Index-Glossary.

```
Library | Hamming code [7,4,3]
Let R be the binary matrix whose columns are the binary vectors of length three
and weight at least two
R=⎛1 1 1 0⎞ : Matrix(ℤ2)
  ⎜1 1 0 1⎟
  ⎝1 0 1 1⎠
The matrices G and H
G=I4|R^T;  H=R|I3;
Encoding function
hamming_encoder(u):=u·G
Decoding function
hamming_decoder(y):=
  begin
    local r, n, s, j
    (r,n)=dimensions(G)
    s=y·H^T
    if not zero?(s) then
      j=index(s, H^T)
      y_j=y_j−1
    end
    take(y, r)
  end
```

```
\caretExample
u=[1, 1, 0, 1] ⟶ [1, 1, 0, 1]
x=hamming_encoder(u) ⟶ [1, 1, 0, 1, 0, 1, 0]
hamming_decoder(x) ⟶ [1, 1, 0, 1]
Let us simulate an error in position 4
e=epsilon_vector(7,4); y=x+e ⟶ [1, 1, 0, 0, 0, 1, 0]
hamming_decoder(y) ⟶ [1, 1, 0, 1]
```

> The notion of an correcting code was introduced by Hamming [in 1950]
>
> V.D. Goppa, [8], p. 46.

Error-reduction factor

Assume that p is the probability that a symbol of T is altered in the transmission. The probability that j errors occur in a block of length n is

$$\binom{n}{j} p^j (1-p)^{n-j}.$$

Therefore

$$P_e(n,t,p) = \sum_{j=t+1}^{n} \binom{n}{j} p^j (1-p)^{n-j} = 1 - \sum_{j=0}^{t} \binom{n}{j} p^j (1-p)^{n-j} \quad [1.1]$$

gives the probability that $t+1$ or more errors occur in a code vector, and this is the probability that the received vector is either undecodable or that an undetectable error occurs.

If we assume that N blocks of k symbols are transmitted, the number of expected errors is pkN if no coding is used, while the expected number of errors if coding is used is at most $(N \cdot P_e(n,t,p)) \cdot k$ (in this product we count as errors all the symbols corresponding to a code error, but of course some of those symbols may in fact be correct). Hence the quotient

$$\rho(n,t,p) = P_e(n,t,p)/p,$$

which we will call the *error reduction factor* of the code, is an upper bound for the average number of errors that will occur in the case of using coding per error produced without coding (cf. E.1.8). For p small enough, $\rho(n,t,p) < 1$ and the closer to 0, the better error correction resulting from the code. The value of $\rho(n,t,p)$ can be computed with the function erf(n,t,p).

1.10 ***Example*** (Error-reduction factor of the Hamming code)**.** According to the formula [1.1], the error-reduction factor of the Hamming code $C \sim [7,4,3]$ for a bit error probability p has the form

$$\binom{7}{2} p(1-p)^5 + \cdots$$

which is of the order $21p$ for p small. Note that this is 7 times the error-reduction factor of the code $\mathit{Rep}(3)$, so that the latter is more effective in reducing errors than C, even if we take into account that the true error reduction factor of C is smaller than $21p$ (because in the error-reduction factor we count all four bits corresponding to a code error as errors, even though some of them may be correct). On the other hand the rates of C and $\mathit{Rep}(3)$ are $4/7$ and $1/3$, respectively, a fact that may lead one to prefer C to $\mathit{Rep}(3)$ under some circumstances.

E.1.8. Let $p' = p'(n,t,p)$ be the probability of a symbol error using a code of length n and correcting capacity t. Let $\bar{p} = P_e(n,t,p)$. Prove that $\bar{p} \geqslant p' \geqslant \bar{p}/k$.

Elementary parameter bounds

In order to obtain efficient coding and decoding schemes with small error probability, it is necessary that k and d are high, inasmuch as the maximum number of errors that the code can correct is $t = \lfloor (d-1)/2 \rfloor$ and that the transmission rate is, given n, proportional to k.

Library | Probabilaty of code error (an upper bound) and error reduction factor

$$P\_e(n,t,p) := \sum_{j=t+1}^{n} \binom{n}{j} \cdot p^j \cdot (1-p)^{n-j}$$

```
erf(n,t,p) := if p=0 then
                0
              else
                P_e(n,t,p)/p
              end
ρ=erf
```

```
n=3; t=1; p=0.01;
P_e(n,t, p), ρ(n,t,p) → 0.000298, 0.0298
n=7; t=1; p=0.01;
P_e(n,t, p), ρ(n,t,p) → 0.002031, 0.2031
n=23; t=3; p=0.01;
P_e(n,t, p), ρ(n,t,p) → 7.6053·10^-5, 0.0076
```

It turns out, however, that k and d cannot be increased independently and arbitrarily in their ranges (for the extreme cases $k = n$ or $d = n$, see E.1.3). In fact, the goal of this section, and of later parts of this chapter, is to establish several non trivial restrictions of those parameters. In practice these restrictions imply, for a given n, that if we want to improve the rate then we will get a lower correcting capability, and conversely, if we want to improve the correcting capability, then the transmission rate will decrease.

Singleton bound and MDS codes

Let us begin with the simplest of the restrictions we will establish.

1.11 Proposition (Singleton bound)**.** *For any code of type* $[n, k, d]$*,*

$$k + d \leqslant n + 1.$$

Proof: Indeed, if C is any code of type (n, M, d), let us write $C' \subseteq T^{n-d+1}$ to denote the subset obtained by discarding the last $d - 1$ symbols of each vector of C. Then C' has the same cardinal as C, by definition of d, and so $q^k = M = |C| = |C'| \leqslant q^{n-d+1}$. Hence $k \leqslant n-d+1$, which is equivalent to the stated inequality. □

MDS codes. Codes that satisfy the equality in the Singleton inequality are called *maximum distance separable* codes, or *MDS codes* for short. The repetition code $\mathsf{Rep}(3)$ and the Hamming code [7,4,3] are MDS codes. The *repetition code* of any length n on the alphabet T, which by definition is $\mathsf{Rep}_q(n) = \{\mathbf{t}_n \,|\, t \in T\}$, is also an MDS code (since it has q elements, its dimension is 1, and it is clear that the distance between any two distinct code-words is n).

Values of $A_2(n,d)$			
n	$d=3$	$d=5$	$d=7$
5	4	2	—
6	8	2	—
7	16	2	2
8	20	4	2
9	40	6	2
10	72–79	12	2
11	144–158	24	4
12	256	32	4
13	512	64	8
14	1024	128	16
15	2048	256	32
16	2720–3276	256–340	36–37

Table 1.1: *Some known values or bounds for* $A_2(n,d)$

1.12 Remark. If C is a code with only one word, then $k_C = 0$, but d_C is undefined. If we want to assign a conventional value to d_C that satisfies the Singleton bound, it has to satisfy $d_C \leqslant n+1$. On the other hand, the Singleton bound tells us that $d_C \leqslant n$ for all codes C such that $|C| \geqslant 2$. This suggests to put $d_C = n+1$ for one-word codes C, as we will do henceforth. With this all one-word codes are MDS codes.

The function $A_q(n,d)$

Given positive integers n and d, let $R_q(n,d)$ denote the maximum of the rates $R_C = k_C/n$ for all codes C of length n and minimum distance d.

We will also set $k_q(n,d)$ and $A_q(n,d)$ to denote the corresponding number of information symbols and the cardinal of the code, respectively, so that $R_q(n,d) = k_q(n,d)/n$ and $A_q(n,d) = q^{k(n,d)}$.

A code C (of length n and mimimum distance d) is said to be *optimal* if $R_C = R_q(n,d)$ or, equivalently, if either $k_C = k_q(n,d)$ or $M_C = A_q(n,d)$.

E.1.9. If $q \geqslant 2$ is an integer, show that $A_q(3,2) = q^2$. In particular we have that $A_2(3,2) = 4$ and $A_3(3,2) = 9$. *Hint:* if $T = \mathbb{Z}_q$, consider the code $C = \{(x,y,x+y) \in T^3 \,|\, x,y \in T\}$.

E.1.10. Show that $A_2(3k,2k) = 4$ for all integers $k \geqslant 1$.

E.1.11. Show that $A_2(5,3) = 4$.

The exact value of $A_q(n,d)$ is unknown in general (this has often been called the *main problem* of coding theory). There are some cases which are

very easy, like $A_q(n,1) = q^n$, $A_2(4,3) = 2$, or the cases considered in E.1.9 and E.1.11. The values $A_2(6,3) = 8$ and $A_2(7,3) = 16$ are also fairily easy to determine (see E.1.19), but most of them require, even for small n, much work and insight. The table 1.1 gives a few values of $A_2(n,d)$, when they are known, and the best known bounds (lower and upper) otherwise. Some of the facts included in the table will be established in this text; for a more comprehensive table, the reader is referred to the table 9.1 on page 248 of the book [6]. It is also interesting to visit the web page

http://www.csl.sony.co.jp/person/morelos/ecc/codes.html

in which there are links to pages that support a dialog for finding the best bounds known for a given pair (n,d), with indications of how they are ascertained.

E.1.12. In the Table 1.1 only values of $A_2(n,d)$ with d odd are included. Show that if d is odd, then $A_2(n,d) = A_2(n+1,d+1)$. Thus the table also gives values for even minimum distance. *Hint:* given a binary code C of length n, consider the code $\overline{C}$ obtained by adding to each vector of C the binary sum of its bits (this is called the *parity completion* of C).

The Hamming upper bound

Before stating and proving the main result of this section we need an auxiliary result.

1.13 Lemma. *Let r be a non-negative integer and $x \in T^n$. Then*

$$|B(x,r)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

Proof: The number of elements of T^n that are at a distance i from an element $x \in T^n$ is

$$\binom{n}{i}(q-1)^i$$

and so

$$|B(x,r)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i,$$

as claimed. □

The lemma shows that the cardinal of $B(x,r)$ only depends on n and r, and not on x, and we shall write $\mathit{vol}_q(n,r)$ to denote it. By the preceeding lemma we have

$$\mathit{vol}_q(n,r) = |B(x,r)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i. \qquad [1.2]$$

Library | Computation of $\mathrm{vol}_q(n,r)$

$$\mathrm{volume}(n,r,q) := \sum_{i=0}^{r} \binom{n}{i} \cdot (q-1)^i$$

$$\mathrm{volume}(n,r) := \sum_{i=0}^{r} \binom{n}{i}$$

\caretvolume(9,3,3) → 835
volume(9,3) → 130

1.14 Theorem (Hamming upper bound)**.** *If $t = \lfloor (d-1)/2 \rfloor$, then the following inequality holds:*

$$A_q(n,d) \leqslant \frac{q^n}{\mathit{vol}_q(n,t)}.$$

Proof: Let C be a code of type $(n, M, d)_q$. Taking into account that the balls of radius $t = \lfloor (d-1)/2 \rfloor$ and center elements of C are pair-wise disjoint (this follows from the definition t and the triangular inequality of the Hamming distance), it turns out that $\sum_{x \in C} |B(x,t)| \leqslant |T^n| = q^n$.

On the other hand we know that

$$|B(x,t)| = \mathit{vol}_q(n,t)$$

and hence

$$q^n \geqslant \sum_{x \in C} |B(x,t)| = M \cdot \mathit{vol}_q(n,t).$$

Now if we take C optimal, we get

$$A_q(n,d)\mathit{vol}_q(n,t) \leqslant q^n,$$

which is equivalent to the inequality in the statement. □

1.15 Remark. The Hamming upper bound is also called *sphere-packing upper bound*, or simply sphere upper bound.

E.1.13. Let m and s be integers such that $1 \leqslant s \leqslant m$ and let $c_1, \ldots, c_m \in \mathbb{F}^n$, where $\mathbb{F}$ is a finite field with q elements. Show that the number of vectors that are linear combinations of at most s vectors from among $c_1, \ldots, c_m$ is bounded above by $\mathit{vol}_q(m,s)$.

[Hamming] established the upper bound for codes

V.D. Goppa, [8], p. 46.

Library | Hamming upper bound (also called sphere upper bound)

$$\text{ub\_sphere}(n,d,q) := \left\lfloor \frac{q^n}{\text{volume}(n, \lfloor (d-1)/2 \rfloor, q)} \right\rfloor$$

$$\text{ub\_sphere}(n,d) := \left\lfloor \frac{2^n}{\text{volume}(n, \lfloor (d-1)/2 \rfloor)} \right\rfloor$$

\caretub_sphere(8,3) → 28
ub_sphere(9,3) → 51
ub_sphere(10,3) → 93
ub_sphere(11,3) → 170
ub_sphere(11,3,3) → 7702
ub_sphere(21,5) → 9039

Perfect codes

In general D_C is a proper subset of T^n, which means that there are elements $y \in T^n$ for which there is no $x \in C$ with $\mathsf{hd}(y,x) \leqslant t$. If $D_C = T^n$, then C is said to be *perfect*. In this case, for every $y \in T^n$ there is a (necessarily unique) $x \in C$ such that $\mathsf{hd}(y,x) \leqslant t$.

Taking into account the reasoning involved in proving the sphere-bound, we see that the necessary and sufficient condition for a code C to be perfect is that

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i = q^n/M \quad (= q^{n-k}),$$

where $M = |C| = q^k$ (this will be called the *sphere* or *perfect* condition).

The total code T^n and the binary repetion code of **odd** length are examples of perfect codes, with parameters $(n, q^n, 1)$ and $(2m+1, 2, 2m+1)$, respectively. Such codes are said to be *trivial perfect codes*. We have also seen that the Hamming code [7,4,3] is perfect (actually this has been checked in Example 1.9).

E.1.14. In next section we will see that if q is a prime-power and r a positive integer, then there are codes with parameters

$$[(q^r-1)/(q-1), (q^r-1)/(q-1) - r, 3]).$$

Check that these parameters satisfy the condition for a perfect code. Note that for $q = 2$ and $r = 3$ we have the parameters [7,4,3].

E.1.15. Show that the parameters $[23, 12, 7]$, $[90, 78, 5]$ and $[11, 6, 5]_3$ satisfy the perfect condition.

E.1.16. Can a perfect code have even minimum distance?

E.1.17. If there is a perfect code of length n and minimum distance d, what is the value of $A_q(n,d)$? What is the value of $A_2(7,3)$?

E.1.18. Consider the binary code consisting of the following 16 words:

0000000	1111111	1000101	1100010
0110001	1011000	0101100	0010110
0001011	0111010	0011101	1001110
0100111	1010011	1101001	1110100

Is it perfect? *Hint:* show that it is equivalent to the Hamming [7,4,3] code.

1.16 Remarks. The parameters of any non-trivial q-ary perfect code, with q a prime power, must be those of a Hamming code, or $[23, 12, 7]$, or $[11, 6, 5]_3$ (van Lint and Tietäväinen (1975); idependently established by Zinovi'ev and Leont'ev (1973)). There are non-linear codes with the same parameters as the Hamming codes (Vasili'ev (1962) for binary codes; Schönheim (1968) and Lindström (1969) in general). Codes with the parameters $(23, 2^{12}, 7)$ and $(11, 3^6, 5)_3$ exist (binary and ternary Golay codes; see Chapter 3), and they are unique up to equivalence (Pless (1968), Delsarte and Goethals (1975); see [25], Theorem 4.3.2, for a rather accessible proof in the binary case, and [11], Chapter 20, for a much more involved proof in the ternary case).

One of the steps along the way of characterizing the parameters of q-ary perfect codes, q a prime power, was to show (van Lint, H.W. Lenstra, A.M. Odlyzko were the main contributors) that the only non-trivial parameters that satisfy the perfect condition are those of the Hamming codes, the Golay codes (see E.1.15, and also P.1.7 for the binary case), and $(90, 2^{78}, 5)_2$. The latter, however, cannot exist (see P.1.8).

Finally let us note that it is conjectured that there are no non-trivial q-ary perfect codes for q not a prime power. There are some partial results that lend support to this conjecture (mainly due to Pless (1982)), but the remaining cases are judged to be very difficult.

The Gilbert inequality

We can also easily obtain a lower bound for $A_q(n, d)$. If C is an optimal code, any element of T^n lies at a distance $\leqslant d-1$ of an element of C, for otherwise there would be a word $y \in T^n$ lying at a distance $\geqslant d$ from all elements of C and $C \cup \{y\}$ would be a code of length n, minimum distance d and with a greater cardinal than |C|, contradicting the optimality of C. This means that the union of the balls of radius $d-1$ and with center the elements of C is the whole T^n. From this it follows that $A_q(n, d) \cdot \mathsf{vol}_q(n, d-1) \geqslant q^n$. Thus we have proved the following:

1.17 Theorem (Gilbert lower bound)**.** *The function $A_q(n, d)$ satisfies the fol-*

lowing inequality:

$$A_q(n,d) \geqslant \frac{q^n}{\mathit{vol}_q(n,d-1)}.$$

What is remarkable about the Gilbert lower bound, with the improvement we will find in the next chapter by means of linear codes, is that it is the only known general lower bound. This is in sharp contrast with the variety of upper bounds that have been discovered and of which the Singleton and sphere upper bounds are just two cases that we have already established.

Library | Gilbert lower bound

$$\mathrm{lb\_gilbert}(n,d,q) := \left\lceil \frac{q^n}{\mathrm{volume}(n,d-1,q)} \right\rceil$$

$$\mathrm{lb\_gilbert}(n,d) := \left\lceil \frac{2^n}{\mathrm{volume}(n,d-1)} \right\rceil$$

\caretlb_gilbert(8,3) → 7
lb_gilbert(9,3) → 12
lb_gilbert(10,3) → 19
lb_gilbert(11,3) → 31
lb_gilbert(11,3,3) → 729
lb_gilbert(21,5) → 278

1.18 Remark. The Hamming and Gilbert bounds are not very close. For example, we have seen that $7 \leqslant A_2(8,3) \leqslant 28$, $12 \leqslant A_2(9,3) \leqslant 51$, $19 \leqslant A_2(10,3) \leqslant 93$ and $31 \leqslant A_2(11,3) \leqslant 170$. But in fact $A_2(8,3) = 20$ (we will get this later, but note that we already have $A_2(8,3) \geqslant 20$ by E.1.4), $A_2(9,3) = 40$ and the best known intervals for the other two are $72 \leqslant A_2(10,3) \leqslant 79$ and $144 \leqslant A_2(11,3) \leqslant 158$ (see Table 1.1).

E.1.19. The sphere upper bound for codes of type $(6,M,3)$ turns out to be $M \leqslant 9$ (check this), but according to the table 1.1 we have $A_2(6,3) = 8$, so that there is no code of type $(6,9,3)$. Prove this. *Hint:* assuming such a code exists, show that it contains three words that have the same symbols in the last two positions.

Summary

- Key general ideas about information generation, coding and transmission.
- The definition of block code and some basic related notions (codewords, dimension, transmission rate, minimum distance and equivalence criteria between codes).

- Abstract decoders and error-correcting capacity.
- The minimum distance decoder and its correcting capacity $t_C = \lfloor (d-1)/2 \rfloor$.
- The Hamming code $[7, 4, 3]$ (our computational approach includes the construction of the code and the coding and decoding functions).
- Error-reduction factor of a code.
- Singleton bound: $k \leqslant n + 1 - d$.
- The function $A_q(n, d)$, whose determination is sometimes called the 'main problem' of block error-correcting codes.Table 1.1 provides some information on $A_2(n, d)$.
- Hamming upper bound:

  $$A_q(n, d) \leqslant q^n / \mathsf{vol}_q(n, t), \text{ where } t = \lfloor (d-1)/2 \rfloor.$$

- Perfect codes: a code $(n, M, d)_q$ is perfect if and only if

  $$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i = q^n/M \quad (= q^{n-k}).$$

- Gilbert lower bound: $A_q(n, d) \geqslant q^n / \mathsf{vol}_q(n, d-1)$.

Problems

P.1.1 (Error-detection and correction)**.** We have seen that a code C of type (n, M, d) can be used to detect up to $d-1$ errors or to correct up to $t = \lfloor (d-1)/2 \rfloor$ errors. Show that C can be used to simultaneously detect up to $s \geqslant t$ errors and correct up to t errors if $t + s < d$.

P.1.2 Prove that $A_2(8, 5) = 4$ and that all codes of type $(8, 4, 5)_2$ are equivalent. *Hint:* by replacing a binary optimal code of length 8 and minimum distance 5 by an equivalent one, we may assume that 00000000 is a code word and then there can be at most one word of weight $\geqslant 6$.

P.1.3. Show that for binary codes of odd minimum distance the Hamming upper bound is not worse than the Singleton upper bound. Is the same true for even minimum distance? And for q-ary codes with $q > 2$?

P.1.4 Prove that $A_2(n, d) \leqslant 2A_2(n-1, d)$. *Hint:* if C is an optimal binary code of length n and minimum distance d, we may assume, changing C into an equivalent code if necessary, that both 0 and 1 appear in the last position of elements of C, and then it is useful to consider, for $i = 0, 1$, the codes $C_i = \{x \in C \mid x_n = i\}$.

P.1.5 (Plotkin construction, 1960)**.** Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be binary codes of types (n, M_1, d_1) and (n, M_2, d_2), respectively. Let $\mathcal{C}$ be the length $2n$ code whose words have the form $x|(x + y)$, for all $x \in \mathcal{C}_1$ and $y \in \mathcal{C}_2$. Prove that $C \sim (2n, M_1M_2, d)$, where $d = \min(2d_1, d_2)$.

P.1.6 Show that $A_2(16, 3) \geqslant 2560$. *Hint:* use Example E.1.4 and the Plotkin construction.

P.1.7 If $C \sim (n, M, 7)$ is a perfect binary code, prove that $n = 7$ or $n = 23$. *Hint:* use the sphere upper bound.

P.1.8. Show that there is no code with parameters $(90, 2^{78}, 5)$. *Hint:* Extracted from [9], proof of Theorem 9.7: if C were a code with those parameters, let X be the set of vectors in C that have weight 5 and begin with two 1s, Y the set of vectors in $\mathbb{Z}_2^{90}$ that have weight 3 and begin with two 1s, and $D = \{(x, y) \in X \times Y \mid S(y) \subset S(x)\}$, and count $|D|$ in two ways ($S(x)$ is the support of x, that is, the set of indices i in $1..90$ such that $x_i = 1$).

Springer

http://www.springer.com/978-3-540-00395-3

Block Error-Correcting Codes
A Computational Primer
Xambo-Descamps, S.
2003, X, 266 p. With online files/update., Softcover
ISBN: 978-3-540-00395-3