Introduction to Coding Theory

Alain Couvreur

November 16, 2020

Chapter 1

Introduction and motivations

Error correcting codes are introduced to preserve the quality of information transmitted across a noisy channel. The classical situation is described as follows:

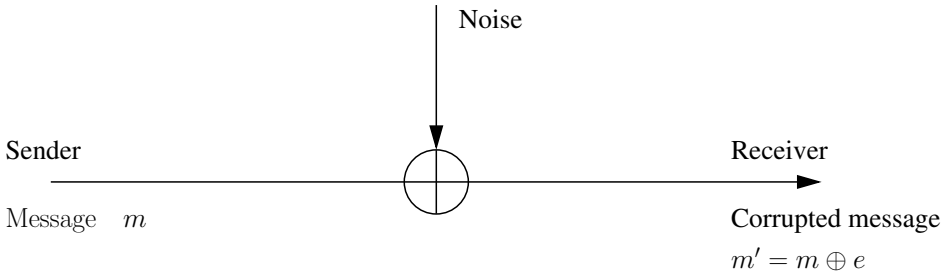


Figure 1.1: A communication channel

Error correcting codes provide a way to reduce the influence of the noise. The principle of error correcting codes consists in adding redundancy in the message so that the receiver could recover the sent message even if it has been corrupted during the transmission. The situation is described in Figure 1.2.

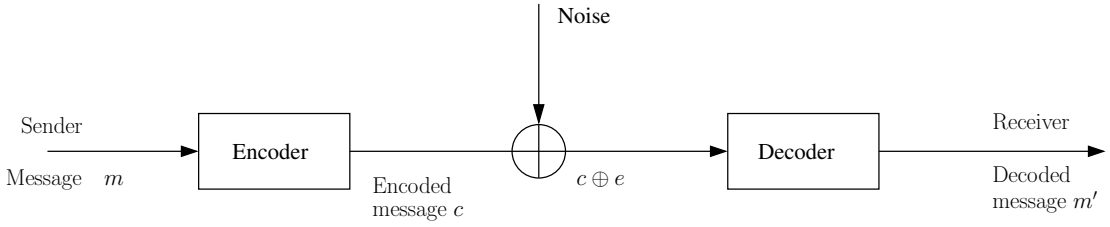


Figure 1.2: Encoding and decoding

The sender wishes to send a message m, then the message is encoded as a message c. The receiver, gets a corrupted message $c \oplus e$, inputs this word in the decoder which returns m'. Our wish is that if the noise is "reasonable", then $m' = m$.

1.1 Error correcting codes

The point of encoding summarizes in one single sentence: encoding consists in adding redundancy in the message.

1.1.1 Tutorial examples

The French social security number

A classical elementary example are French social security numbers. These numbers consist of a sequence of 13 digits together with two additional digits. For instance:

$$2\ 77\ 04\ 59\ 606\ 122 — 61$$

The first 13 digits correspond to information on the owner of the number:

- 2 means that the owner is a woman (1 for men);
- 77 is its year of birth (namely 1977);
- 04, means that she is born in April;
- 59 is the number of the department of her birth (*Département du Nord*)
- 606 is the town where she is born (Valenciennes)
- 122 means that she's the 122nd person[1] born in Valenciennes in April 1977.

The digits 61 do not give additional information on the owner they are obtained by an elementary mathematical process:

$$61 \equiv -2770459696122 \mod 97.$$

It is equal to the opposite of the remainder of the 13 digits number (regarded as an integer in base 10) by the division by 97, represented as a integer in the range $\{1, \ldots, 97\}$.

This is an example of **error detecting code**. Basically, if you enter your social security number on the health insurance website and make an error, this error will be detected unless the error on the 13 digits does not change the remainder modulo 97.

[1] If the number of births in a month exceeds 999 in a town, then, the counter restarts from 001 and another number is given for this town this number will describe the town only for this particular month.

International Standard Book Number

This example is taken in J. Walkers book [Wal00]. The International Standard Book Number (ISBN) is an international number identifying every edition of a published book. It consists of a sequence of 10 digits[2]. The first 9 digits provide information on the book while the 10th one is a redundancy symbol obtained as follows. For instance, consider the book *A course in Error Correcting Codes*, written by J. Justesen and T. Høholdt [JH04]. The ISBN number of the book is

$$3\ 03719\ 001\ 9$$

and the final 9 is obtained as

$$9 \equiv 3 \times 1 + 0 \times 2 + 3 \times 3 + 7 \times 4 + 1 \times 5 + 9 \times 6 + 0 \times 7 + 0 \times 8 + 1 \times 9 \mod 11.$$

That is, for each of the first digit multiply it by its position number (from 1 to 9), then sum up all these products and take the remainder of the division by 11. If this remainder is equal to 10, then the final symbol is an X. This yields another example of error detecting code.

Error correcting codes are based on the very same principle consisting in adding redundancy to information, in order to detect and possibly to correct errors in a corrupted message.

1.2 Error correcting codes, basic notions

In this course, we only consider the case of block codes: the message is first decomposed in blocks of bits of fixed length k, that is to say in vectors in $\mathbb{F}_2^k$. Let us consider from now on, that our message consists of a single block $m \in \mathbb{F}_2^k$. An encoding map is an injective map $\mathbb{F}_2^k \longrightarrow \mathbb{F}_2^n$ for some integer $n > k$.

For instance the maps

$$\rho : \left\{ \begin{array}{ccc} \mathbb{F}_2 & \longrightarrow & \mathbb{F}_2^5 \\ (b) & \longmapsto & (bbbbb) \end{array} \right.$$

and

$$\pi : \left\{ \begin{array}{ccc} \mathbb{F}_2^7 & \longrightarrow & \mathbb{F}_2^8 \\ (b_0, b_1, b_2, b_3, b_4, b_5, b_6) & \longmapsto & (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7) \end{array} \right. ,$$

where $b_7 = \sum_{i=0}^{6} b_i$ are encoding maps. In terms of error detection and correction, the first encoding map allows to correct up to 2 errors by a simple majority voting. For instance if the received word is (01011), one can reasonably hope that the sender sent a 1. The second encoding map does not allow error correction but detects the presence of one error. Indeed, it is easy to see that the image of π consists of words having an even number of 1's. Thus, if the receiver gets a word with an odd number of 1's, then he can directly conclude to the presence of at least one error in the message (and more generally of an odd number of

[2]Since January 1st 2007, it has been extended to a 13-digit sequence since the former system became insufficient due to the growth of the number of published books.

errors). Notice, that the receiver is completely unable to detect the presence of 2 errors or more generally of an even number of errors.

Notice that the previous considerations do not concern directly the encoding maps but their images, thus from now on, we will focus on *error correcting codes*, which are defined as the image of an encoding map. Before giving formal definitions, let us finish the present introduction with a few remarks:

- In this course the encoding maps will always be linear and hence, codes will be vector spaces. It should be noticed that there exists a theory of nonlinear codes but we will not discuss it in the present course.
- The two examples above considered encoding maps defined on fixed length bit strings that is on the space $\mathbb{F}_2^k$ for some integer k, this seems natural for every computer theorist, on the other hand, for many reasons appearing in what follows, it is extremely relevant to consider a more general context and consider codes defined over a general finite field $\mathbb{F}_q$ where q is some prime power.

1.2.1 Codes and their parameters

First of all, even if the definition has been sketched before, let us state a formal definition of an error correcting code.

Definition 1.1 (Code, length, dimension)**.** Let n be a positive integer, a *linear error correcting code* $\mathscr{C}$ is a vector subspace of $\mathbb{F}_q^n$. The integer n is called the *length* of $\mathscr{C}$. The *dimension* of $\mathscr{C}$ is its dimension as an $\mathbb{F}_q$–vector space and is in general denoted by k:

$$k = \dim_{\mathbb{F}_q} \mathscr{C}.$$

Hamming distance and minimum distance

Definition 1.2. Given an element $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$, the Hamming weight of x is defined as

$$\mathrm{w_H}(\boldsymbol{x}) \stackrel{\text{def}}{=} |\{i \mid x_i \neq 0\}|.$$

The Hamming distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$\mathrm{d_H}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \mathrm{w_H}(\boldsymbol{x} - \boldsymbol{y}).$$

A *ball* or *Hamming ball* of center $\boldsymbol{x} \in \mathbb{F}_q^m$ and radius $r \leqslant n$ is a ball for the Hamming distance, that is

$$\mathbf{B}_{\mathrm{H}}(\boldsymbol{x}, r) \stackrel{\text{def}}{=} \{\boldsymbol{y} \in \mathbb{F}_q^n \mid \mathrm{d_H}(\boldsymbol{x}, \boldsymbol{y}) \leqslant r\}.$$

Remark 1. One can prove that $\mathrm{d_H}$ is an actual distance, i.e. it is symmetric, zero if and only if $\boldsymbol{x} = \boldsymbol{y}$ and satisfies the triangle inequality. The verification is left as an exercise.

Definition 1.3 (Minimum distance of a code)**.** Let $\mathscr{C}$ be a linear code of length n. The *minimum distance of* $\mathscr{C}$ is the minimum distance between two distinct codewords of $\mathscr{C}$. That is:

$$\mathrm{d}_{\min}(\mathscr{C}) \stackrel{\text{def}}{=} \min_{\boldsymbol{x},\boldsymbol{y}\in\mathscr{C},\ \boldsymbol{x}\neq\boldsymbol{y}} \{\mathrm{d}_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})\}. \tag{1.1}$$

By linearity, it can equivalently be defined as

$$\mathrm{d}_{\min}(\mathscr{C}) = \min_{\boldsymbol{x}\in\mathscr{C}\setminus\{0\}} \{\mathrm{w}_{\mathrm{H}}(\boldsymbol{x})\}. \tag{1.2}$$

The minimum distance of a code is in general denoted as d.

The minimum distance quantifies the theoretical capability of error correction. In particular, the following elementary lemma asserts that, for a code of minimum distance d if a received word has less than $\frac{d-1}{2}$ errors, then it can be corrected. We stress here that this is purely theoretical, in particular, we do not say that this decoding can be performed easily, the existence of an efficient decoding algorithm is far from being guaranteed. This will be discussed in the next chapter.

Lemma 1.4. *Let $\mathscr{C}$ be a code of minimum distance d, then the balls $\mathbf{B}_H\left(\boldsymbol{c}, \lfloor\frac{d-1}{2}\rfloor\right)$ for $\boldsymbol{c}\in\mathscr{C}$ are pairwise disjoint, that is*

$$\forall \boldsymbol{c},\boldsymbol{c}'\in\mathscr{C}, \boldsymbol{c}\neq\boldsymbol{c}',\ \mathbf{B}_H\left(\boldsymbol{c}, \left\lfloor\frac{d-1}{2}\right\rfloor\right) \cap \mathbf{B}_H\left(\boldsymbol{c}', \left\lfloor\frac{d-1}{2}\right\rfloor\right) = \emptyset.$$

Proof. Let $\boldsymbol{c},\boldsymbol{c}'$ be two distinct words in $\mathscr{C}$. Assume that there exists $\boldsymbol{x}$ in $\mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}, \lfloor\frac{d-1}{2}\rfloor\right)\big) \cap \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}', \lfloor\frac{d-1}{2}\rfloor\right)$. Then the triangle inequality asserts that

$$\begin{aligned}\mathrm{d}_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{c}') &\leqslant \mathrm{d}_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{x}) + \mathrm{d}_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{c}')\\ &\leqslant \left\lfloor\frac{d-1}{2}\right\rfloor + \left\lfloor\frac{d-1}{2}\right\rfloor\\ &\leqslant d-1.\end{aligned}$$

But, since $\mathscr{C}$ has minimum distance d, then this contradicts the assumption $\boldsymbol{c}\neq\boldsymbol{c}'$. □

Summary and discussion on the parameters

As explained, to a linear code we associate three parameters:

- the length n, which is the length of the blocks (and the dimension of the ambient space);
- the dimension k, which is its dimension as an $\mathbb{F}_q$–vector space;
- and the minimum distance d.

Notation 1.1. From now on, we use the notation "$\mathscr{C}$ is an $[n,k,d]_q$ code" to say "$\mathscr{C}$ is a code of length n, dimension k and minimum distance d over $\mathbb{F}_q$. We also speak about $[n,k]_q$ codes for codes of length n and dimension k.

Before starting a discussion on the parameters, let us introduce, the relative parameters.

Definition 1.5 (Relative parameters)**.** Given a code $\mathscr{C}$ of length n, dimension k and minimum distance d, the *rate* of $\mathscr{C}$ is defined as

$$R \stackrel{\text{def}}{=} \frac{k}{n}$$

and the *relative distance* as

$$\delta \stackrel{\text{def}}{=} \frac{d}{n}.$$

The rate and relative distance are rational numbers in $[0,1]$. The rate quantifies the efficiency of the code. It is the ratio between information bits and sent bits. A rate close to 0 corresponds to a very redundant code which requires a huge amount of energy to transmit a short message. A rate close to 1 corresponds to an efficient code for which the ratio of pure information in the transmitted bit string is close to 1. On the other hand, the relative distance quantifies the theoretical capability to correct errors. The closer δ to 1, the larger number of errors one can theoretically correct.

Clearly, our objective is that both the rate and the relative distance are close to 1. Unfortunately, these requirements are in contradiction. Indeed, as we will see in the next chapters, there are several upper bounds implying the impossibility to have both the rate and the relative distance close to 1. The most famous one is the so-called *Singleton bound* which asserts that

$$R + \delta \leqslant 1 + \frac{1}{n}.$$

Thus, a "good code" will be a code satisfying a good trade off between these two relative parameters.

Another conclusion of this observation is that, one cannot have an efficient (i.e. with low redundancy) encoding and correct many errors. Thus the choice of codes will depend on the situation where they are used: for instance if the channel is very noisy, we will probably choose a code with a large minimum distance, even if its rate is low. On the other hand some devices require a limitation of energy consumption, and hence will encourage to use a code of high rate. Notice that many other facts should be taken into account. For instance, in some communications, one can ask the sender to resend a corrupted block, in such a situation, if this "re-send operation" is easy to perform, then one can choose a high rate code. On the other hand, this "re-send operation" may be impossible in long-distance communications, for instance with spacecrafts. As we will see in the chapter on Reed Muller codes, NASA used a $[32,6,16]_2$ code to receive photos of Mars from the spacecraft *Mariner IX*. Using this code, NASA could correct up to 7 errors per block while the rate is rather low (0.375). Even if the spacecraft had limited memory and energy resources, it was important to be able to correct a large number of errors, since there was no possible interaction with the spacecraft and hence it was not possible to ask it to resend some corrupted photo.

The designed parameters

It is frequent that, for a given code, the exact parameters are unknown but that lower bounds for them are known. In this situation, these lower bounds are called the *designed parameters*. Notice that in general, the dimension is known or can be computed by Gaussian elimination. On the other hand it will be noticed further that the computation of the minimum distance of a code is a hard algorithmic problem. Thus, it is frequent to deal with codes whose actual minimum distance is unknown, while a lower bound, i.e. a *designed minimum distance* is known. This designed distance is fundamental since in general, the decoding algorithms will correct errors as soon as their number is less than half the designed distance and not up to half the actual distance (which is unknown).

Nonlinear codes

One can more generally define an error correcting code as a subset $\mathscr{C} \subseteq \mathbb{F}_q^n$. For non linear codes, one can still define a minimum distance using (1.1) but be careful not using (1.2) which is in general irrelevant[3] for nonlinear codes. Instead of dealing with the dimension which cannot be defined if the code is not a vector space, one can consider the number of codewords, which is frequently denoted as M. Then a natural analog for the dimension is $\log_q(M)$.

While the theory of nonlinear codes is rich and subject to many interesting developments, we mostly deal with linear codes in this course. The use and the interest of linear codes will be motivated in Section 1.2.2. From now on, the term *code* will always mean *linear code*.

1.2.2 How to describe a code?

There are two manners to describe a code, which are the two classical manners to describe a vector subspace of $\mathbb{F}_q^n$. Namely, one can either give a basis or at least a family of generators or give a system of linear equations whose solution space is the code. More formally, a code can be represented either as the image of some matrix or as the kernel of another matrix. This motivates the following definitions.

Definition 1.6 (Generator matrix)**.** Let $\mathscr{C}$ be an $[n, k, d]_q$ code. A *generator matrix* of $\mathscr{C}$ is a matrix $\boldsymbol{G} \in \mathfrak{M}_{\ell,n}(\mathbb{F}_q)$ for some $\ell \geqslant k$ whose rows form a family of generators of $\mathscr{C}$. That is

$$\mathscr{C} = \left\{ \boldsymbol{m}\boldsymbol{G} \mid \boldsymbol{m} \in \mathbb{F}_q^{\ell} \right\}.$$

Definition 1.7 (Parity-check matrix)**.** Let $\mathscr{C}$ be an $[n, k, d]_q$ code. A *parity-check matrix* of C is a matrix $\boldsymbol{H} \in \mathfrak{M}_{n-\ell,n}(\mathbb{F}_q)$ for some $\ell \geqslant k$ whose right kernel equals $\mathscr{C}$. That is

$$\mathscr{C} = \{\boldsymbol{x} \in \mathbb{F}_q^n \mid \boldsymbol{H}\boldsymbol{x}^T = 0\}.$$

[3]Actually, for (1.2) to be relevant, the code only needs to be *additive* (i.e. the code must be an additive group), which is weaker than being linear (unless the base field is $\mathbb{F}_2$).

Remark 2. Frequently in the literature, a generator matrix is defined as an $k \times n$ matrix whose rows form a basis of $\mathscr{C}$ and a parity-check matrix is defined as an $(n-k) \times n$ matrix whose right kernel is $\mathscr{C}$. For many reasons which will appear in what follows, we chose this more general definition. We will speak of *full rank generator matrix* (resp. *full rank parity–check matrix*) when the matrix has k (resp. $n-k$) rows.

Remark 3. It is worth noting that one cannot say "the generator matrix of $\mathscr{C}$" but "a generator matrix of $\mathscr{C}$". Indeed, if $\boldsymbol{G} \in \mathfrak{M}_{\ell,n}(\mathbb{F}_q)$ is a generator matrix of $\mathscr{C}$, then for all invertible $\ell \times \ell$ matrix $\mathbf{S}$, then the matrix $\mathbf{S}\boldsymbol{G}$ is another generator matrix. Thus, such a matrix is not unique. For the very same reason, a parity–check matrix is not unique either.

A motivation for using linear codes

This description is actually the main motivation for using linear codes instead of nonlinear ones. Indeed, generator or parity–check matrices provide a very "compact" description of a code. To compare, if we have to describe a nonlinear code, then we need to list all of its codewords. If this code is binary (defined over $\mathbb{F}_2$) and contains M words, then we need nM bits to describe it completely. On the other hand, if the code is linear then, the representation by a generator matrix requires only $nk = n\log_2(M)$ bits. Thus, the memory size necessary to store a nonlinear code of M words is exponentially larger than the size necessary to store a linear code with the same number of words.

Systematic codes

Definition 1.8 (Systematic generator matrix)**.** Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $\boldsymbol{G} \in \mathfrak{M}_{k,n}(\mathbb{F}_q)$ be a full–rank generator matrix of $\mathscr{C}$. The matrix $\boldsymbol{G}$ is said to be *systematic* if it is of the form

$$\boldsymbol{G} = \left(\begin{array}{c|c} I_k & A \end{array} \right)$$

for some matrix $A \in \mathfrak{M}_{k,n-k}(\mathbb{F}_q)$. A code is said to be *systematic* if one of its generator matrix is systematic.

Not any code is systematic. In particular, the following lemma characterizes such codes.

Lemma 1.9. *A code $\mathscr{C} \subseteq \mathbb{F}_q^n$ is systematic if an only if for any full–rank generator matrix $\boldsymbol{G}$ of $\mathscr{C}$ the k first columns of $\boldsymbol{G}$ are linearly independent, or equivalently the $k \times k$ minor composed by the k most left-hand columns of $\boldsymbol{G}$ is nonzero.*

Proof. First note that if $\boldsymbol{G}$ and $\boldsymbol{G}'$ are two generator matrices for $\mathscr{C}$, then, there exists $\mathbf{P} \in \mathbf{GL}(k, \mathbb{F}_q)$ such that $\boldsymbol{G} = \mathbf{P}\boldsymbol{G}'$. Therefore if the most left–hand minor of $\boldsymbol{G}$ is nonzero, then so is that of $\boldsymbol{G}'$.

If $\mathscr{C}$ is systematic, then it has a systematic generator matrix whose k first columns are obviously independent since they form the canonical basis of $\mathbb{F}_q^k$. Conversely, if $\mathscr{C}$ has a generator matrix whose k first columns are linearly independent, then, by performing Gaussian elimination on $\boldsymbol{G}$, we get a systematic generator matrix for $\mathscr{C}$. □

Remark 4. Given a code $\mathscr{C}$ with a full–rank generator matrix $\boldsymbol{G} \in \mathfrak{M}_{k,n}(\mathbb{F}_q)$, the map

$$\left\{ \begin{array}{ccc} \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ \mathbf{m} & \longmapsto & \mathbf{m} \cdot \boldsymbol{G} \end{array} \right.$$

is an encoding map as introduced in § 1.2. If $\boldsymbol{G}$ is systematic, then the corresponding encoding map is nothing but a map consisting to append to the original message $\mathbf{m} \in \mathbb{F}_q^k$ a redundancy block of length $n - k$.

Another interest of systematic generator matrix if exist is that they are unique.

Lemma 1.10. *Let $\mathscr{C}$ be a systematic code. Then there is a unique generator matrix for $\mathscr{C}$ in systematic form.*

Proof. Let $\boldsymbol{G}, \boldsymbol{G}'$ be two generator matrices of $\mathscr{C}$ in systematic form. Since they are both generator matrices for $\mathscr{C}$, there exists $\mathbf{P} \in \mathbf{GL}(k, \mathbb{F}_q)$ such that

$$\boldsymbol{G} = \mathbf{P} \cdot \boldsymbol{G}'. \tag{1.3}$$

Since both matrices have an I_k as left–hand block, (1.3) entails $\mathbf{P} = I_k$. □

Parity–check matrices and the minimum distance

An important property of a parity–check matrix is that the minimum distance of the code "can be read" by studying the linear relations between the columns of the matrix. The following elementary lemma is frequently very useful in coding theory.

Lemma 1.11. *Let $\mathscr{C}$ be a code of length n and minimum distance d. Let $\boldsymbol{H}$ be a parity–check matrix of $\mathscr{C}$, then every $(d-1)$–tuple of columns of $\boldsymbol{H}$ are linearly independent and there is at least one linearly linked d–tuple of columns.*

Proof. Denote by $H_1, \ldots, H_n$ the columns of the matrix $\boldsymbol{H}$. Let $\boldsymbol{c} \in \mathscr{C}$ be a codeword of weight w. Let $i_1, \ldots, i_w$ be the indexes in $\{1, \ldots, n\}$ such that $c_i \neq 0$ if and only if $i \in \{i_1, \ldots, i_w\}$. The relation

$$\boldsymbol{H}\boldsymbol{c}^T = 0$$

is equivalent to

$$c_{i_1} H_{i_1} + \cdots + c_{i_w} H_{i_w} = 0.$$

That is to say: zero linear combinations of w distinct columns of $\boldsymbol{H}$ are in one-to-one correspondence with codewords of weight w in $\mathscr{C}$. This yields the result. □

Corollary 1.12. *Let $\mathscr{C}$ be a code with parity–check matrix $\boldsymbol{H}$. Let d be the minimum distance of $\mathscr{C}$.*

(i) If $\boldsymbol{H}$ has no zero column, then, $d > 1$.

(ii) If the columns of $\boldsymbol{H}$ are pairwise non collinear, then $d > 2$.

1.3 First examples

1.3.1 The repetition code

The most naive way to add redundancy to data is to repeat the message several times. If the alphabet we use is $\mathbb{F}_2$, then a 5-time repetition encoding map sends the bit 0 onto the word (00000) and the bit 1 onto the word (11111)

The corresponding code is represented by a generator matrix $\boldsymbol{G}$ and a parity–check matrix $\boldsymbol{H}$ defined as follows

$$\boldsymbol{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{H} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}.$$

More generally, for any positive integer n and any finite field, one can define the repetition code as the code with a generator matrix $\boldsymbol{G} \in \mathfrak{M}_{1,n}(\mathbb{F}_q)$ of the form $(1\ 1\ \cdots\ 1)$. Such a code has length n, dimension 1 and minimum distance n. Indeed, the nonzero codewords of such a code are of the form $(a\ a\ \cdots\ a)$ for $a \in \mathbb{F}_q^\times$.

Summary of the particularities of this code:

- It is $[n, 1, n]_q$.
- Its rate is $R_n = \frac{1}{n}$, in particular, $\lim_{n\to+\infty} R_n = 0$.
- For each block, one can correct up to $\lfloor \frac{n-1}{2} \rfloor$ errors. Indeed, given a corrupted received word $\boldsymbol{y}$ with less than $\lfloor \frac{n-1}{2} \rfloor$ errors, find the unique element $a \in \mathbb{F}_q$ such that the majority of digits of $\boldsymbol{y}$ are equal to a; then the original word is equal to $(a\ a\ \cdots\ a)$.

As a conclusion, this code has a very bad rate (asymptotically zero) but has a very good error correction capacity.

1.3.2 The parity code

The parity code is the image of the following encoding map.

$$\begin{cases} \mathbb{F}_q^{n-1} & \longrightarrow & \mathbb{F}_q^n \\ (x_1, \ldots, x_{n-1}) & \longmapsto & (x_1, \ldots, x_{n-1}, -\sum_{i=1}^{n-1} x_i) \end{cases}$$

The corresponding code has generator and parity–check matrices defined as follows:

$$\boldsymbol{G} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{H} = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \end{pmatrix}.$$

If $q = 2$, then the code is the set of words whose number of 1's is even, this is the rationale behind the terminology *parity code*.

The properties of this code are:

- Its parameters are $[n, n-1, 2]$;
- In particular, its relative distance $\delta_n = \frac{2}{n}$ and hence $\lim_{n\to+\infty} \delta_n = 0$.
- The code is not error correcting, but it is error detecting: it can detect the presence of one error.

Remark 5. The parity code is practically used in the RAID (*Redundant array of independent disks*) system. RAID system consists in distributing data in several hard drive and adding some redundant data so that the data remains recoverable even after a failure of one or several hard drives.

- RAID 1 is based on the principle of the repetition code: the data is contained in n hard drives, each one contains a mirror copy of the other one.
- RAID 5 is based on the principle of the parity code: data is distributed on $n-1$ hard drives while the n-th one is a parity disk : its i-th bit is the binary sum of the i-th bits of the $n-1$ other disks.

Remark 6. You may have noticed that a generator matrix of the repetition code is a parity–check matrix of the parity code and conversely. We will see in the chapter on duality that these codes are actually dual to each other.

1.3.3 The Hamming code

This is the first non trivial construction of codes. For an integer $\ell \geqslant 3$, a Hamming code is a binary code defined by an $\ell \times (2^\ell - 1)$ parity–check matrix $\boldsymbol{H}_\ell$ whose columns are pairwise distinct and list all nonzero words of $\mathbb{F}_2^\ell$. For instance, if $\ell = 3$, the code with parity–check matrix $\boldsymbol{H}_3$ and generator matrix $\boldsymbol{G}_3$ defined as

$$\boldsymbol{H}_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{G}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

is a Hamming code.

Proposition 1.13. *A Hamming code has parameters* $[2^\ell - 1, 2^\ell - 1 - \ell, 3]$.

Proof. The length is clear. For the dimension, the code has an $\ell \times (2^\ell - 1)$ parity–check matrix. To prove that the code has dimension $2^\ell - 1 - \ell$, we only have to prove that this matrix has full rank. It is true since every nonzero word in $\mathbb{F}_2^\ell$ is a column of $\boldsymbol{H}_\ell$, one can extract ℓ columns of $\boldsymbol{H}_\ell$ which form a basis of $\mathbb{F}_2^\ell$, which proves that $\boldsymbol{H}_\ell$ has rank ℓ. Thus the corresponding code has dimension $2^\ell - 1 - \ell$.

Finally, for the minimum distance we can use Lemma 1.11 and Corollary 1.12. More precisely, it is clear that $\boldsymbol{H}_\ell$ has no zero column (by definition) and that every two columns

are distinct and hence non collinear[4]. Thus, from Corollary 1.12, the minimum distance is at least 3. Second, one sees easily that there are triples of columns which are linearly linked: since the set of columns of $\boldsymbol{H}_\ell$ is that of nonzero words in $\mathbb{F}_2^\ell$, given two distinct columns H_i, H_j of $\boldsymbol{H}_\ell$, the column $H_i + H_j$ is also a column of $\boldsymbol{H}_\ell$. From Lemma 1.11, this proves that the minimum distance of the code is exactly 3. □

Remark 7. One could wonder why we chose $\ell \geqslant 3$ in the definition. Actually, one can take $\ell = 2$ and give this definition but this gives the repetition code of length 3. Details are left to the reader.

Hamming Codes are perfect These codes have a very particular and rare property: they are *perfect*. Let us first give the definition of a perfect code.

Definition 1.14. A code $\mathscr{C}$ of parameters $[n, k, d]_q$ is said to be *perfect* if

$$\mathbb{F}_q^n = \bigcup_{\boldsymbol{c}\in\mathscr{C}} \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}, \left\lfloor \frac{d-1}{2} \right\rfloor\right).$$

Lemma 1.15. *Hamming codes are perfect.*

Proof. Since the codes have minimum distance 3, we have $\left\lfloor \frac{d-1}{2} \right\rfloor = 1$. Moreover, the cardinality of a Hamming ball of radius 1 is $n + 1 = 2^\ell$ since a word in a Hamming ball centred at a word $\boldsymbol{c}$ is either c or a word obtained from $\boldsymbol{c}$ after flipping one bit. Now, if we compute the total volume of the disjoint union of radius 1 balls centred at codewords, we get

$$\sum_{\boldsymbol{c}\in\mathscr{C}} \left|\mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}, \left\lfloor \frac{d-1}{2} \right\rfloor\right)\right| = \sum_{\boldsymbol{c}\in\mathscr{C}} 2^\ell = 2^{2^\ell-1-\ell} \times 2^\ell = 2^{2^\ell-1},$$

which is nothing but the total number of elements of the ambient space $\mathbb{F}_2^{2^\ell-1}$ □

Notice that, from Lemma 1.4, the balls centred at codewords and of radius $\left\lfloor \frac{d-1}{2} \right\rfloor$ are pairwise distinct. The point is that, in general they are far to cover the whole space. But it holds for the very particular case of perfect codes. A direct consequence is that for perfect codes, there exists a decoding procedure[5] returning a unique codeword for any input vector.

Hamming codes are 1–correcting One can easily correct one error with such a code. Let us explain how in the special case $\ell = 3$. Let $\mathscr{C}$ be a Hamming code for $\ell = 3$. It is a $[7, 4, 3]$ code. From now on, denote by $\boldsymbol{H}$ (instead of $\boldsymbol{H}_3$) a parity–check matrix of $\mathscr{C}$. Let $\boldsymbol{c} \in \mathscr{C}$ and $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ be a corrupted codeword where $\boldsymbol{e}$ has weight 1. That is $\boldsymbol{y}$ is a codeword with one error. Moreover, $\boldsymbol{e}$ is some vector of the canonical basis of $\mathbb{F}_2^7$. Since $\boldsymbol{H}c^T = 0$, we have

$$\boldsymbol{H}\boldsymbol{y}^T = \boldsymbol{H}(\boldsymbol{c} + \boldsymbol{e})^T = \boldsymbol{H}\boldsymbol{e}^T.$$

Finally, if $\boldsymbol{e}$ is the i–th vector of the canonical basis of $\mathbb{F}_2^7$ for some $1 \leqslant i \leqslant 7$. Then $\boldsymbol{H}\boldsymbol{e}^T$ is nothing but the i–th column of $\boldsymbol{H}$. This provides a simple decoding algorithm for the code.

[4]Over $\mathbb{F}_2$, two nonzero words are collinear if and only if they are equal.

[5]Caution: we never said that this decoding procedure is algorithmically efficient. It is purely theoretic.

Algorithm 1: A decoding algorithm for a $[7,4,3]_2$ Hamming code correcting one error

Input : A corrupted word $\boldsymbol{y} \in \mathbb{F}_2^7$.
Output: A codeword $\boldsymbol{c} \in \mathscr{C}$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ for some word $\boldsymbol{e}$ of weight 1.

$\boldsymbol{s} \leftarrow \boldsymbol{H}\boldsymbol{y}^T$;
Find $i \in \{1, \ldots, 7\}$ such that $\boldsymbol{s}$ equals the i–th column of $\boldsymbol{H}$;
Let $\boldsymbol{e}_i$ be the i–th vector of the canonical basis;
return $\boldsymbol{y} + \boldsymbol{e}_i$;

1.4 Constructing new codes from old

In this section we describe several operations to transform a code to another one.

1.4.1 Extended codes

For a binary code of length n with at least one word of odd weight, one can add append any codeword with a parity bit (sometimes called *check sum*) in order to get a new code of length $n+1$ whose words have all even weight. This notion can be defined for a longer code

Definition 1.16. Let $\mathscr{C} \in \mathbb{F}_q^n$ be a code with at least one element $\boldsymbol{c} \in \mathscr{C}$ such that $c_1 + \cdots + c_n \neq 0$. The *extended* code of $\mathscr{C}$ is the code

$$\mathrm{Ext}(\mathscr{C}) \stackrel{\text{def}}{=} \left\{ (c_1, \ldots, c_n, -\sum_{i=1}^{n} c_i) \;\middle|\; \boldsymbol{c} = (c_1, \ldots, c_n) \in \mathscr{C} \right\}$$

The words of the extended code satisfy the property that the sum of their digits is always zero.

Proposition 1.17. *Let $\mathscr{C}$ be a code of length n and $\boldsymbol{H} \in \mathfrak{M}_{\ell,n}(\mathbb{F}_q)$ be a parity check matrix for $\mathscr{C}$. Then, the $(\ell+1) \times (n+1)$ matrix*

$$\boldsymbol{H}' \stackrel{def}{=} \begin{pmatrix} & & & 0 \\ & \boldsymbol{H} & & \vdots \\ & & & 0 \\ 1 & \ldots & 1 & 1 \end{pmatrix}$$

is a parity–check matrix for $\mathrm{Ext}(\mathscr{C})$.

Proof. It is a straightforward verification. □

1.4.2 Shortening and puncturing

The most elementary operation to get a short code from a longer one is probably puncturing which only consists in deleting some prescribed entries of any codeword.

Definition 1.18 (Puncturing)**.** Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $I \subseteq \{1, \dots, n\}$. Then the *puncturing* $\mathcal{P}_I(\mathscr{C})$ of $\mathscr{C}$ at I is a code of length $n - |I|$ obtained as follows

$$\mathcal{P}_I(\mathscr{C}) = \{(c_i)_{i \in \{1,\dots,n\} \setminus I} \mid \boldsymbol{c} \in \mathscr{C}\}.$$

It is the set of codewords of $\mathscr{C}$ whose i–th entry has been deleted for any $i \in I$.

Lemma 1.19. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code of dimension k. Let $\boldsymbol{G} \in \mathfrak{M}_{\ell,n}(\mathbb{F}_q)$ be a generator matrix of $\mathscr{C}$. Let $I \subseteq \{1, \dots, n\}$. Then the matrix $\boldsymbol{G}'$ obtained from $\boldsymbol{G}$ by deleting the columns whose index is in I is a generator matrix for $\mathcal{P}_I(\mathscr{C})$.*

Proof. Straightforward. □

Proposition 1.20. *Let $\mathscr{C}$ be an $[n, k, d]_q$ code and $I \subseteq \{1, \dots, n\}$. Then, then code $\mathcal{P}_I(\mathscr{C})$ is $[n', k', d']$ with*

$$\begin{aligned} n' &= n - |I| \\ k' &\leqslant k \\ d - |I| \leqslant d' &\leqslant d. \end{aligned}$$

Proof. The statement for the length is obvious.

For the dimension, consider a full–rank generator matrix $\boldsymbol{G} \in \mathfrak{M}_{k,n}(\mathbb{F}_q)$ for $\mathscr{C}$. From Lemma 1.19 we get a generator matrix $\boldsymbol{G}' \in \mathfrak{M}_{k,n-|I|}(\mathbb{F}_q)$ for $\mathcal{P}_I(\mathscr{C})$. Since this matrix may fail to have full rank, we can only assert that $k' \leqslant k$.

For the minimum distance, for any word $\boldsymbol{c} \in \mathscr{C}$, the word $\mathcal{P}_I(\mathscr{C}) \stackrel{\text{def}}{=} (c_i)_{i \in \{1,\dots,n\} \setminus I}$ has weight

$$\mathrm{w_H}(\mathcal{P}_I(\boldsymbol{c})) \geqslant \mathrm{w_H}(\boldsymbol{c}) - |I|. \tag{1.4}$$

Equality holds if I is entirely contained in the support of $\boldsymbol{c}$, i.e. if for any $i \in I$, $c_i \neq 0$. On the other hand, we obviously have

$$\mathrm{w_H}(\mathcal{P}_I(\boldsymbol{c})) \leqslant \mathrm{w_H}(\boldsymbol{c}). \tag{1.5}$$

Considering the minimum over all the nonzero words of $\mathcal{P}_I(\mathscr{C})$ of (1.4) and (1.5), we get the statement for the minimum distance. □

Another construction, which is in some sense dual to puncturing (see Chapter 5) is shortening.

Definition 1.21 (Shortening)**.** Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $I \subseteq \{1, \dots, n\}$. Then the *shortening* $\mathcal{S}_I(\mathscr{C})$ of $\mathscr{C}$ at I is a code of length $n - |I|$ obtained as follows

$$\mathcal{S}_I(\mathscr{C}) = \{(c_i)_{i \in \{1,\dots,n\} \setminus I} \mid \boldsymbol{c} \in \mathscr{C}, \text{ and, } \forall j \in I,\ c_j = 0\}.$$

It is the set of codewords of $\mathscr{C}$ whose i–th entry is zero for all $i \in I$ where these entries have been deleted.

Let us start with an obvious fact.

Lemma 1.22. *Let $\boldsymbol{c}' \in \mathcal{S}_I(\mathscr{C})$, then there exists a codeword $\boldsymbol{c} \in \mathscr{C}$ such that*

$$\forall i \in \{1, \dots, n\}, \quad c_i = \begin{cases} c'_i & \text{if} \quad i \notin I \\ 0 & \text{if} \quad i \in I. \end{cases}$$

Lemma 1.23. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $\boldsymbol{H} \in \mathfrak{M}_{\ell,n}(\mathbb{F}_q)$ be a parity–check matrix of $\mathscr{C}$. Let $I \subseteq \{1, \dots, n\}$, then the matrix $\boldsymbol{H}'$ obtained from $\boldsymbol{H}$ by deleting the columns whose index is in I is a parity–check matrix for $\mathcal{S}_I(\mathscr{C})$.*

Proof. Since $\boldsymbol{c} \in \mathscr{C}$, by definition of $\boldsymbol{H}$, we have $\boldsymbol{H} \cdot \boldsymbol{c}^T = 0$ and hence it is straightforward to check that $\boldsymbol{H}' \cdot \boldsymbol{c}' = 0$. Hence, we proved that $\mathcal{S}_I(\mathscr{C}) \subseteq \ker \boldsymbol{H}'$.

Conversely, let $\boldsymbol{c}' \in \mathbb{F}_q^{n-|I|}$ be such that $\boldsymbol{H}' \cdot \boldsymbol{c}'^T = 0$. Then, construct the word $\boldsymbol{c} \in \mathbb{F}_q^n$ as in Lemma 1.22. Since $\boldsymbol{H}' \cdot \boldsymbol{c}'^T = 0$ we get $\boldsymbol{H} \cdot \boldsymbol{c} = 0$ and hence $\boldsymbol{c} \in \mathscr{C}$. Thus, $\boldsymbol{c}' \in \mathcal{S}_I(\mathscr{C})$. Thus, $\ker \boldsymbol{H} = \mathcal{S}_I(\mathscr{C})$, which concludes the proof. □

Proposition 1.24. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be an $[n, k, d]_q$ code and $I \subseteq \{1, \dots, n\}$. Then $\mathcal{S}_I(\mathscr{C})$ is $[n', k', d']$ with*

$$\begin{aligned} n' &= n - |I| \\ k' &\geqslant k - |I| \\ d' &\geqslant d. \end{aligned}$$

Proof. The statement on the length is obvious.

For the dimension consider a full rank parity–check matrix $\boldsymbol{H} \in \mathfrak{M}_{n-k,n}(\mathbb{F}_q)$ and consider the parity–check matrix $\boldsymbol{H}' \in \mathfrak{M}_{k,n-|I|}(\mathbb{F}_q)$ of $\mathcal{S}_I(\mathscr{C})$ constructed as in Lemma 1.23. Then, since $\boldsymbol{H}'$ may fail to be full rank we can only assert that $\dim k' \geqslant k - |I|$.

For the minimum distance, notice that for any $\boldsymbol{c}' \in \mathcal{S}_I(\mathscr{C})$ there exists $\boldsymbol{c} \in \mathscr{C}$ as in Lemma 1.22 and obviously,

$$\mathrm{w_H}(\boldsymbol{c}') = \mathrm{w_H}(\boldsymbol{c}).$$

By taking the minimum of the above equation over all $\boldsymbol{c}' \in \mathcal{S}_I(\mathscr{C}) \setminus \{0\}$ we get the result. □

1.4.3 Subfield subcode, trace code

Another manner to construct a code from another one is to change the base field. Some constructions of codes can be done only over a large enough base field, this is for instance a drawback of Reed Solomon codes (see Chapter 6). On the other hand for many practical applications it is preferable to have a code defined over a small field: ideally $\mathbb{F}_2$.

In what follows, m denotes an integer larger than 1. There exists two manners to construct a code over a subfield from a code over a larger field. The first one is subfield subcode.

Definition 1.25 (Subfield subcode)**.** Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$. The *subfield subcode* of $\mathscr{C}$ over $\mathbb{F}_q$ is denote by $\mathscr{C}_{|\mathbb{F}_q}$ and defined as

$$\mathscr{C}_{|\mathbb{F}_q} \stackrel{\text{def}}{=} \mathscr{C} \cap \mathbb{F}_q^n.$$

Proposition 1.26. *If $\mathscr{C}$ is an $[n, n-r, d]_{q^m}$ code, then $\mathscr{C}_{|\mathbb{F}_q}$ is an $[n, \geqslant n-mr, \geqslant d]_q$ code.*

Proof. $\mathscr{C}_{|\mathbb{F}_q}$ is contained in $\mathscr{C}$ hence its minimum distance is at least as large as that of $\mathscr{C}$. For the dimension, Consider the $\mathbb{F}_q$–linear map

$$\phi : \left\{ \begin{array}{ccc} \mathbb{F}_{q^m}^n & \longrightarrow & \mathbb{F}_{q^m}^n \\ (x_1, \ldots, x_n) & \longmapsto & (x_1^q - x_1, \ldots, x_n^q - x_n) \end{array} \right. .$$

The kernel of ϕ is $\mathbb{F}_q^n$ which has $\mathbb{F}_q$–dimension n. Thus, $\mathrm{Im}(\phi)$ has $\mathbb{F}_q$–dimension $n(m-1)$. Now consider the restriction $\phi_{|\mathscr{C}} : \mathscr{C} \to \mathbb{F}_{q^m}^n$. Its image has $\mathbb{F}_{q^m}^n$–dimension at most $n(m-1)$ and hence

$$\begin{aligned} \dim_{\mathbb{F}_q} \ker \phi_{|\mathscr{C}} &\geqslant \dim_{\mathbb{F}_q} \mathscr{C} - n(m-1) \\ &\geqslant m(n-r) - n(m-1) \\ &\geqslant n - mr. \end{aligned}$$

Moreover, $\ker \phi_{|\mathscr{C}}$ is nothing but $\mathscr{C}_{|\mathbb{F}_q}$. □

The second way to construct a code over a subfield is the trace construction. Its definition requires the definition of the trace map over finite fields which we recall here.

Definition 1.27. Let $\mathbb{F}_q$ be a finite field and $\mathbb{F}_{q^m}$ be an extension. Let $a \in \mathbb{F}_{q^m}$, the trace of a over $\mathbb{F}_q$ is denoted by $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a)$ and defined as

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a) \stackrel{\text{def}}{=} a + a^q + \cdots + a^{q^{m-1}}.$$

The trace of an element of $\mathbb{F}_{q^m}$ is in $\mathbb{F}_q$. Indeed, one checks easily that for all $a \in \mathbb{F}_{q^m}$,

$$\mathrm{Tr}(a)^q = (a + a^q + \cdots + a^{q^{m-1}})^q = a^q + a^{q^2} + \cdots + a^{q^m}.$$

Then, since $a \in \mathbb{F}_{q^m}$, $a^{q^m} = a$ and we deduce that $\mathrm{Tr}(a)^q = \mathrm{Tr}(a)$ which entails that $\mathrm{Tr}(a) \in \mathbb{F}_q$.

Remark 8. The terminology *trace* is explained as follows. Consider the map given by the multiplication by a:

$$\left\{ \begin{array}{ccc} \mathbb{F}_{q^m} & \longrightarrow & \mathbb{F}_{q^m} \\ x & \longmapsto & a \cdot x. \end{array} \right.$$

Regarding this map as an $\mathbb{F}_q$–linear endomorphism of an $\mathbb{F}_q$–linear space of dimension m. Then the trace of this endomorphism, i.e. the trace of any matrix representation of this endomorphism, is nothing but $\mathrm{Tr}(a)$.

Lemma 1.28. *The trace $Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q} : \mathbb{F}_{q^m} \to \mathbb{F}_q$ is $\mathbb{F}_q$–linear and surjective.*

Proof. The $\mathbb{F}_q$–linearity is a consequence of the $\mathbb{F}_q$–linearity of the Frobenius map $x \mapsto x^q$. Therefore, the trace map is an $\mathbb{F}_q$–linear form on $\mathbb{F}_{q^m}$ regarded as an $\mathbb{F}_q$-vector space. Since a linear form has its image contained in a vector space of dimension 1, it is either zero or surjective. Thus, let us prove that the trace map is nonzero. Indeed, the kernel of $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ is the set of elements $a \in \mathbb{F}_{q^m}$ such that

$$a + a^q + \cdots + a^{q^{m-1}} \neq 0.$$

Thus, it is the set of roots of the polynomial $X + X^q + \cdots + X^{q^{m-1}}$. Since this polynomial is nonzero it has at most q^{m-1} roots and hence cannot vanish on the whole $\mathbb{F}_{q^m}$. This concludes the proof. □

Corollary 1.29. *Let $a \in \mathbb{F}_{q^m}$ such that for any $\lambda \in \mathbb{F}_{q^m}$ we have $Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\lambda a) = 0$, then $a = 0$.*

Proof. If a was nonzero then for any $b \in \mathbb{F}_{q^m}$, we would have

$$\mathrm{Tr}(b) = \mathrm{Tr}(aa^{-1}b)$$

which would be 0 by assumption on a. Thus, the trace map would be zero, which contradicts its surjectivity. □

Now we have the material to define trace codes and study some of their properties.

Definition 1.30 (Trace code)**.** Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$. The *trace code* of $\mathscr{C}$ over $\mathbb{F}_q$ is defined as

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\mathscr{C}) \stackrel{\text{def}}{=} \left\{ (\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(c_1), \ldots, \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(c_n)) \mid \boldsymbol{c} = (c_1, \ldots, c_n) \in \mathscr{C} \right\}.$$

Remark 9. For convenience sake, when there is no possible ambiguity on the subfield we note preferently $\mathrm{Tr}(\mathscr{C})$ instead of the (rather heavy) notation $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\mathscr{C})$.

Proposition 1.31. *Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$ be a code of $\mathbb{F}_{q^m}$–dimension k. Then, $Tr(\mathscr{C}) \subseteq \mathbb{F}_q^n$ is a code of $\mathbb{F}_q$–dimension at most mk.*

Proof. Consider the map

$$\left\{ \begin{array}{ccc} \mathscr{C} & \longrightarrow & \mathbb{F}_q^n \\ \boldsymbol{c} = (c_1, \ldots, c_n) & \longmapsto & (\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(c_1), \ldots, \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(c_n)) \end{array} \right. .$$

This is an $\mathbb{F}_q$–linear map from $\mathscr{C}$ which, regarded as an $\mathbb{F}_q$–vector space has $\mathbb{F}_q$–dimension mk. The trace code is nothing but the image of this map and hence has $\mathbb{F}_q$–dimension less than or equal to mk. □

Remark 10. In general, no relation exists between the minimum distance of a code and that of its trace code. Indeed, one could have a codeword $\boldsymbol{c} \in \mathscr{C}$ of weight n such that any entry of $\boldsymbol{c}$ has a zero trace. Thus, the trace of $\boldsymbol{c}$ would have weight 1 while $\boldsymbol{c}$ had weight n.

Generator and parity-check matrices of trace codes Actually the generator matrix of a trace code $\mathrm{Tr}(\mathscr{C})$ can easily be deduced from that of $\mathscr{C}$. To explain this explicit expression we need the notion of *dual basis*.

Proposition 1.32 (Dual basis)**.** *Let $(\alpha_1, \ldots, \alpha_m)$ be an $\mathbb{F}_q$–basis of $\mathbb{F}_{q^m}$, then there exists a unique basis $(\alpha_1^*, \ldots, \alpha_m^*)$ called* dual basis *satisfying:*

$$\forall i, j \in \{1, \ldots, n\}, \quad Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha_i \alpha_j^*) = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{else.} \end{cases}$$

Proof. Let us prove the existence of α_1^*. Let $x = x_1\alpha_1 + \cdots + x_n\alpha_n \in \mathbb{F}_{q^m}$ where the x_i's are elements of $\mathbb{F}_q$. Then, consider the system of equations

$$\begin{cases} \mathrm{Tr}(\alpha_2 x) &= 0 \\ \mathrm{Tr}(\alpha_3 x) &= 0 \\ \quad \vdots & \\ \mathrm{Tr}(\alpha_n x) &= 0 \end{cases} \quad \Longrightarrow \quad \begin{cases} \mathrm{Tr}(\alpha_2\alpha_1)x_1 + \mathrm{Tr}(\alpha_2^2)x_2 + \cdots + \mathrm{Tr}(\alpha_2\alpha_n)x_n &= 0 \\ \mathrm{Tr}(\alpha_3\alpha_1)x_1 + \mathrm{Tr}(\alpha_3\alpha_2)x_2 + \cdots + \mathrm{Tr}(\alpha_3\alpha_n)x_n &= 0 \\ \vdots & \\ \mathrm{Tr}(\alpha_n\alpha_1)x_1 + \mathrm{Tr}(\alpha_n\alpha_2)x_2 + \cdots + \mathrm{Tr}(\alpha_n\alpha_n)x_n &= 0 \end{cases}$$

This system has $n-1$ equations and n unkowns $(x_1, \ldots, x_n)$ and hence has a nonzero solution. Let a_1 be such a solution, then we claim that $\mathrm{Tr}(\alpha_1 a_1) \neq 0$. Indeed, if $\mathrm{Tr}(\alpha_1 a_1)$ was 0, since we also have $\mathrm{Tr}(\alpha_i a_1) = 0$ for any $i \geqslant 2$, then $\mathrm{Tr}(\alpha_i a_1)$ would be zero for any i and by the $\mathbb{F}_q$–linearity of the trace, we would have $\mathrm{Tr}(y a_1) = 0$ for any $y \in \mathbb{F}_{q^m}$ which, from Corollary 1.29 is impossible since $a_1 \neq 0$. Therefore, since $\mathrm{Tr}(\alpha_1 a_1) \neq 0$, then set

$$\alpha_1^* \stackrel{\text{def}}{=} \frac{1}{\mathrm{Tr}(\alpha_1 a_1)} \cdot a_1.$$

The α_i^*'s for $i \geqslant 2$ are obtained in the very same manner. There remains to show that it is a basis and that it is unique. To prove it is a basis, let $x_1, \ldots, x_m \in \mathbb{F}_q$ such that

$$x_1\alpha_1^* + \cdots + x_n\alpha_n^* = 0.$$

Then, for any $i \in \{1, \ldots, n\}$ one can apply $\mathrm{Tr}(\alpha_i \cdot\ )$ to this previous equation an, using the definition of the α_i^*'s we get $x_i = 0$. Thus, the family $(\alpha_1^*, \ldots, \alpha_m^*)$ is composed by linearly independent elements, since it has m elements in a space of dimension m, it is a basis.

Finally, let us prove the uniqueness of such a basis. Suppose there exists two elements α_1^* and α_1^{**} such that

$$\mathrm{Tr}(\alpha_1\alpha_1^*) = \mathrm{Tr}(\alpha_1\alpha_1^{**}) = 1 \quad \text{and} \quad \forall i \in \{1, \ldots, m\},\ \mathrm{Tr}(\alpha_i\alpha_1^*) = \mathrm{Tr}(\alpha_i\alpha_1^{**}) = 0.$$

Then, for all $i \in \{1, \ldots, n\}$, $\mathrm{Tr}(\alpha_i(\alpha_1^* - \alpha_1^{**}) = 0$. Thus, by linearity, for any $y \in \mathbb{F}_{q^m}$, $\mathrm{Tr}(y(\alpha_1^* - \alpha_1^{**})) = 0$ which, from Corollary 1.29, entails $\alpha_1^* = \alpha_1^{**}$. □

Remark 11. It is actually a more general result in bilinear algebra: given a non degenerated bilinear map, any basis has a dual basis. Here, the $\mathbb{F}_q$–bilinear map

$$\begin{cases} \mathbb{F}_{q^m} \times \mathbb{F}_{q^m} & \longrightarrow & \mathbb{F}_q \\ (x, y) & \longmapsto & \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(xy) \end{cases}$$

is non degenerate as asserted by Corollary 1.29,

Corollary 1.33. *Let $(\alpha_1, \ldots, \alpha_m)$ be an $\mathbb{F}_q$–basis of $\mathbb{F}_{q^m}$. Let $x \in \mathbb{F}_{q^m}$, then x can be expressed as a linear combination of the α_i's as follows:*

$$x = Tr(\alpha_1^* x)\alpha_1 + \cdots + Tr(\alpha_m^* x)\alpha_m.$$

Proof. There is a unique decomposition of the form

$$x = x_1\alpha_1 + \cdots + x_m\alpha_m$$

for $x_1, \ldots x_m \in \mathbb{F}_q$. Next, let $i \in \{1, \ldots, m\}$, the we have

$$\mathrm{Tr}(\alpha_i^* x) = x_1\mathrm{Tr}(\alpha_i^*\alpha_1) + \cdots + x_n\mathrm{Tr}(\alpha_i^*\alpha_n) = x_i.$$

This concludes the proof. □

In a similar manner any vector $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$ expresses as

$$\boldsymbol{c} = \alpha_1\mathrm{Tr}(\alpha_1^*\boldsymbol{c}) + \cdots \alpha_m\mathrm{Tr}(\alpha_m^*\boldsymbol{c})$$

and we have the following result.

Proposition 1.34. *Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$ be an $\mathbb{F}_{q^m}$–linear code. Let $(\alpha_1, \ldots, \alpha_m)$ be an $\mathbb{F}_q$–basis of $\mathbb{F}_{q^m}$. Let $\boldsymbol{G} \in \mathfrak{M}_{k,n}(\mathbb{F}_{q^m})$ be a generator matrix of $\mathscr{C}$. Denote by $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k$ the rows of $\boldsymbol{G}$ and consider the matrix $\boldsymbol{G}' \in \mathfrak{M}_{mk,n}(\mathbb{F}_q)$ whose rows are $Tr(\alpha_1^*\boldsymbol{r}_1), \ldots Tr(\alpha_m^*\boldsymbol{r}_1)$, $Tr(\alpha_1^*\boldsymbol{r}_2), \ldots Tr(\alpha_m^*\boldsymbol{r}_2), \ldots,$ $Tr(\alpha_1^*\boldsymbol{r}_k), \ldots Tr(\alpha_m^*\boldsymbol{r}_k)$. Then, $\boldsymbol{G}'$ is a generator matrix of $Tr(\mathscr{C})$.*

Before proving the result, let us explain how useful it is. Choose an $\mathbb{F}_q$–basis $(\alpha_1, \ldots, \alpha_m)$ of $\mathbb{F}_{q^m}$. From a generator matrix $\boldsymbol{G}$, express any entry g_{ij} in the basis $(\alpha_1, \ldots, \alpha_m)$. According to Corollary 1.33:

$$g_{ij} = \gamma_1\alpha_1 + \cdots + \gamma_m\alpha_m.$$

Then replace each entry of the matrix by the column: $\begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_m \end{pmatrix}$, and you get a generator matrix for the trace code.

Example 1.35. Consider the finite field $\mathbb{F}_4$ defined as $\mathbb{F}_2[\alpha]$ with α such that $\alpha^2+\alpha+1 = 0$. Over $\mathbb{F}_4$, the code with generator matrix

$$\begin{pmatrix} 0 & 1 & \alpha & \alpha+1 & 0 & 1 \\ \alpha & 1 & \alpha+1 & \alpha & 1 & 0 \end{pmatrix}$$

Then its trace code over $\mathbb{F}_2$ has a generator matrix of the form

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Proof of Proposition 1.34. The rows $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k$ provide an $\mathbb{F}_{q^m}$-basis of $\mathscr{C}$. Then, an $\mathbb{F}_q$–base can be obtained as follows.

$$\alpha_1^* \boldsymbol{r}_1, \ldots, \alpha_m^* \boldsymbol{r}_1, \ldots, \alpha_1^* \boldsymbol{r}_k, \ldots, \alpha_m^* \boldsymbol{r}_k.$$

Since $\mathrm{Tr}(\mathscr{C})$ is nothing but the image of $\mathscr{C}$ by the $\mathbb{F}_q$–linear map $\mathrm{Tr}(\cdot)$, it is spanned over $\mathbb{F}_q$ by an image of an $\mathbb{F}_q$–basis by $\mathrm{Tr}(\cdot)$. This concludes the proof. □

Chapter 2

Decoding problems

In the previous chapter we discussed the first definitions and properties of linear codes without considering the main question, namely *How to use codes to correct errors?* For that, let us formalize the notion of *decoder* or *decoding algorithm* and the various versions of decoding problems.

Definition 2.1 (Decoder)**.** Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be an error correcting code. A *decoder* for $\mathscr{C}$ is a function $\mathcal{D} : \mathbb{F}_q^n \to \mathscr{C} \cup \{?\}$, such that for all $\boldsymbol{c} \in \mathscr{C}$, $\mathcal{D}(\boldsymbol{c}) = \boldsymbol{c}$.

Remark 12. A decoder cannot correct any error pattern and may fail. This is the reason why the decoder may return "?".

The two major features expected from a decoder are:

- A decoder should correct *many* errors, i.e. it should solve some specific decoding problem. See § 2.1 for some examples of decoding problems.
- A decoder should be efficient in terms of time and space complexity. By *efficient* we mean in general a space and time complexity which is polynomial in the code length n.

2.1 Deterministic decoding problems

2.1.1 Examples of decoding problems

Here we list several classical decoding problems.

The bounded decoding problem

Given a code $\mathscr{C} \subseteq \mathbb{F}_q^m$, an integer r and a vector $\boldsymbol{y} \in \mathbb{F}_q^n$, find (if exists) a word $\boldsymbol{c} \in \mathscr{C}$ such that

$$\mathrm{d_H}(\boldsymbol{c}, \boldsymbol{y}) \leqslant r.$$

Remark 13. A close related decision problem is to decide whether there exists a word $\boldsymbol{c} \in \mathscr{C}$ such that $\mathrm{d_H}(\boldsymbol{c}, \boldsymbol{y}) \leqslant r$.

The unambiguous decoding problem

Given a code $\mathscr{C} \subseteq \mathbb{F}_q^m$ and a vector $\boldsymbol{y} \in \mathbb{F}_q^n$, find a word $\boldsymbol{c} \in \mathscr{C}$ such that

$$\mathrm{d_H}(\boldsymbol{c}, \boldsymbol{y}) \leqslant \left\lfloor \frac{d-1}{2} \right\rfloor,$$

where d is the minimum distance of $\mathscr{C}$.

Remark 14. Thanks to Lemma 1.4, the solution of the unambiguous decoding problem, if exists, is unique. Note that to state the problem, one needs to know the minimum distance of the code. However, the minimum distance of a code is difficult to compute in general (see § 2.1.2).

The list decoding problem

Given a code $\mathscr{C} \subseteq \mathbb{F}_q^m$, an integer r and a vector $\boldsymbol{y} \in \mathbb{F}_q^n$, return (if exists) the whole list of words $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_s \in \mathscr{C}$ such that

$$\forall i \in \{1, \ldots, s\}, \quad \mathrm{d_H}(\boldsymbol{c}_i, \boldsymbol{y}) \leqslant r.$$

Remark 15. Decoders as defined in Definition 2.1 cannot solve the list decoding problem in general and one needs to introduct a notion of *list decoder* as a function $\mathbb{F}_q^n \to \mathcal{P}(\mathscr{C}) \cup \{?\}$ where $\mathcal{P}(\mathscr{C})$ denotes the set of subsets of $\mathscr{C}$.

Remark 16. For a list decoding algorithm to be polynomial, the returned list should have polynomial size. In Chapter 8 we discuss further the list decoding problem and state an upper bound called *Johnson bound* for r which asserts that the list has a size polynomial in the code length.

2.1.2 Hardness of decoding

One of the major difficulties of coding theory is that for almost every code, no efficient (i.e. with polynomial space and time complexity) decoder is known. Actually, it has been proved in [BMvT78] by Berlekamp, McEliece and VanTilborg that the decision version of the bounded decoding problem (see Remark 13) is NP–complete.

Similarly, the determination of the minimum distance is a difficult problem in general: more precisely, given a code $\mathscr{C} \subseteq \mathbb{F}_q^n$ and an integer $r \leqslant n$. Deciding whether the minimum distance of $\mathscr{C}$ is less than r is an NP–complete problem.

The hardness of the decoding problem motivated R.J. McEliece [McE78] to apply these problems to cryptography and design a puiblic key encryption scheme based on the hardness of decoding.

2.2 Probabilistic decoding problems

The previous decoding problems require the decoder to correct any error patter with Hamming weight below some threshold.

Another class of decoding problem is probabilistic and consists, for some probabilistic error model in correcting error with a failure probability below some threshold. This problem will be investigated further in Chapter 3. To introduce this approach, we first need a probabilistic modelisation of errors. This is the notion of channel.

2.2.1 Channels

A channel is a theoretical model to describe a communication with possible errors. In this course we will only consider *memoryless* channels. That is, given a transmitted bit or digit the errors corrupting it do not depend on the previous transmitted bits/digits.

The binary symmetric channel

The first and most classical example of channel in coding theory is the so-called *Binary Symmetric Channel* denoted as BSC(p), where $p \in [0, 1]$ is a real parameter. This channel works as follows. If a 0 is sent on the channel, then the receiver gets a 0 with probability $1 - p$ and a 1 with probability p. Conversely, if a 1 is sent, then the receiver get a 1 with probability $1 - p$ and a 0 with probability p. This channel is usually represented by Figure 2.1. The rigorous description of this channel is : let e be a Bernouilli random variable of

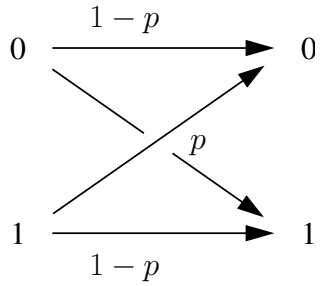


Figure 2.1: The binary symmetric channel

parameter p such that for every transmitted bit b, the receiver gets $b + e$, where the "+" stands for the addition in $\mathbb{F}_2$ (or equivalently the Xor gate).

Remark 17. Actually, one can always assume that $p \leqslant 1/2$. Indeed, if $p > 1/2$, then, after flipping every bit of the received word, we can do as if the message was transmitted across a binary symmetric channel of parameter $1 - p \leqslant 1/2$.

Thanks to the above remark, from now on, whenever, we consider a BSC of parameter p, this parameter is always assumed to be $\leqslant 1/2$.

The binary erasure channel

In the present chapter we mainly deal with the binary symmetric channel. This channel corresponds to a case where during a transmission, some bits may be flipped. Actually in some situations, bits are not flipped but only lost or destroyed. This is represented by the so-called *binary erasure channel* which is described by Figure 2.2.

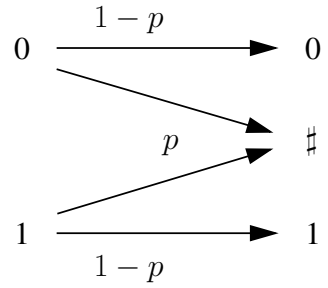


Figure 2.2: The binary erasure channel

The q–ary symmetric and erasure channels

In case of transmission of elements of a finite field $\mathbb{F}_q$ instead of $\mathbb{F}_2$, we need to define another channel called the *q–ary symmetric channel* which is defined as follows. For an input $a \in \mathbb{F}_q$, the receiver gets a with probability $1-p$ or every element $b \in \mathbb{F}_q \setminus \{a\}$ can be received with probability $\frac{p}{q-1}$.

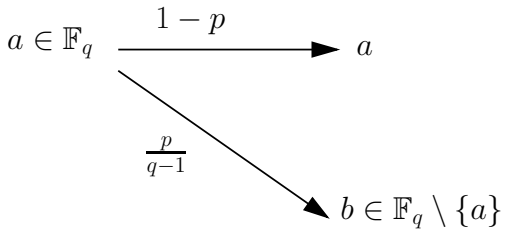


Figure 2.3: The q–ary symmetric channel

In a similar fashion, one can of course define a q–ary erasure channel sending an element $\alpha \in \mathbb{F}_q$ onto itself with probability p and onto $\sharp$ with probability $1-p$.

2.2.2 A probabilistic decoding problem

The problem can be stated as follows. Consider a fixed channel, a constant $\varepsilon > 0$ and a given code $\mathscr{C} \subseteq \mathbb{F}_q^n$.

Problem. Find a decoder $\mathcal{D}$ such that for a uniformly random element $\boldsymbol{c} \in \mathscr{C}$ and a random error pattern $\boldsymbol{e}$ produced by the channel we have

$$\mathbb{P}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c}) < \varepsilon.$$

Compared to the decoding problems presented in § 2.1 for which the goal was to correct any error pattern of weight less than some upper bound, here we aim at correcting "almost any" error pattern produced by the channel. The decoder may fail with some probability (which is expected to be low).

2.2.3 The maximum likelihood decoding problem

Given a code $\mathscr{C}$, a uniformly random vector $\boldsymbol{c} \in \mathscr{C}$ and an error pattern $\boldsymbol{e}$ produced by the channel. Set $\boldsymbol{y} \stackrel{\text{def}}{=} \boldsymbol{c}+\boldsymbol{e}$. Find, (if unique) the vector $\boldsymbol{c}' \in \mathscr{C}$ maximizing the conditional probability

$$\mathbb{P}_{\boldsymbol{e}\sim\text{channel}}(\boldsymbol{c}' \text{ is sent} \mid \boldsymbol{y} \text{ is received}).$$

Relation with the Hamming distance

If the channel is the binary or q–ary symmetric channel of parameter p, then solving the maximum likelihood problem is equivalent to solve the following problem

Problem. Given a code $\mathscr{C} \subset \mathbb{F}_q^n$ and a vector $\boldsymbol{y} \in \mathbb{F}_q^n$, find (if unique) the vector $\boldsymbol{c} \in \mathscr{C}$ such that $\mathrm{d_H}(\boldsymbol{c},\boldsymbol{y}) = \min_{\boldsymbol{x}\in\mathscr{C}} \mathrm{d_H}(\boldsymbol{x},\boldsymbol{y})$.

Indeed, let $\boldsymbol{x}$ be a binary vector lying in a code $\mathscr{C}$ and $\boldsymbol{y} = \boldsymbol{x}+\boldsymbol{e}$ be the vector received after transmission of $\boldsymbol{x}$ across a BSC(p). That is, $\boldsymbol{e} = (e_1,\ldots,e_n)$ is a vector whose entries are independent Bernouilli random variables with parameter p. Then, one sees easily that

$$\mathbb{P}_{\boldsymbol{e}\sim\mathrm{BSC}(p)}(\boldsymbol{x} \text{ is sent} \mid \boldsymbol{y} \text{ is received}) = p^{\mathrm{d_H}(\boldsymbol{x},\boldsymbol{y})}(1-p)^{n-\mathrm{d_H}(\boldsymbol{x},\boldsymbol{y})}.$$

Thus, a solution of the maximum likelihood decoding problem is nothing but a word of $\mathscr{C}$ which is the closest possible to $\boldsymbol{y}$ with respect to the Hamming distance.

2.3 Some example of decoders

2.3.1 The exhaustive decoder

The most elementary decoding algorithm consists in enumerating all the vectors of the code and returns the closest one if unique.

Algorithm 2: The exhaustive decoding alorithm

Input : A code $\mathscr{C} \subseteq \mathbb{F}_q^n$, a word $\boldsymbol{y} \in \mathbb{F}_q^n$.
Output: A codeword $\boldsymbol{c} \in \mathscr{C}$ such that $\mathrm{d_H}(\boldsymbol{y}, \boldsymbol{c}) = \min_{\boldsymbol{x} \in \mathscr{C}}\{\mathrm{d_H}(\boldsymbol{x}, \boldsymbol{y})\}$.

```
dist = n + 1;
tmp = "?";
for x ∈ C do
    if d_H(x, y) < dist then
        tmp = x;
        dist = d_H(x, y);
    end
    if d_H(x, y) = dist then
        tmp = "?";
    end
end
return tmp;
```

This decoder provides a solution for the maximum likelihood decoding problem (if the channel is the binary symmetric channel). Unfortunately, unless for very short and small dimensional codes this decoder cannot be used in practice because of its time complexity which is $O(q^k)$ and hence is exponential in the code dimension.

2.3.2 The syndrome decoder

The syndrome decoder is a generalization of the decoding algorithm proposed for the $[7, 4, 3]_2$ Hamming code in § 1.3.3.

Definition 2.2. Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $\boldsymbol{H} \in \mathfrak{M}_{n-k,n}(\mathbb{F}_q)$ be a full rank parity check matrix of $\mathscr{C}$. Let $\boldsymbol{y} \in \mathbb{F}_q^n$. The *syndrome* of $\boldsymbol{y}$ with respect to $\boldsymbol{H}$ is defined as

$$S(\boldsymbol{y}) \stackrel{\text{def}}{=} \boldsymbol{H} \cdot \boldsymbol{y}^T.$$

Lemma 2.3. *Let $\boldsymbol{e} \in \mathbb{F}_q^m$ be an error pattern and $\boldsymbol{c} \in \mathscr{C}$. Set $\boldsymbol{y} \stackrel{def}{=} \boldsymbol{c} + \boldsymbol{e}$. Then the syndrome of $\boldsymbol{y}$ depends only on $\boldsymbol{e}$. Namely,*

$$S(\boldsymbol{y}) = S(\boldsymbol{e}).$$

Proof. This is a straightforward consequence of the definition of a parity check matrix. □

Proposition 2.4. *Let $\mathscr{C}$ be a code of minimum distance d with a parity–check matrix $\boldsymbol{H}$. The syndromes of vectors $\boldsymbol{e}$ of weight $\leqslant \frac{d-1}{2}$ are pairwise distinct.*

Proof. Assume that $\boldsymbol{e}, \boldsymbol{e}'$ both have weight $\leqslant \frac{d-1}{2}$ and have the same syndrome, i.e. $S(\boldsymbol{e}) = S(\boldsymbol{e}')$. Then $\mathrm{w_H}(\boldsymbol{e} - \boldsymbol{e}') \leqslant d - 1$ and

$$S(\boldsymbol{e} - \boldsymbol{e}') = \boldsymbol{H} \cdot (\boldsymbol{e} - \boldsymbol{e}') = S(\boldsymbol{e}) - S(\boldsymbol{e}') = 0.$$

Therefore $\boldsymbol{e} - \boldsymbol{e}' \in \mathscr{C}$ and has weight less than $d - 1$ which contradicts the fact that the minimum distance is d. □

Here is the principle of syndrome decoding. Construct a dictionary containing any pair $(S(\boldsymbol{e}), \boldsymbol{e})$ for all $\boldsymbol{e}$ of weight less than or equal to $\frac{d-1}{2}$. Use a data structure so that given an entry $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, the pair $(S(\boldsymbol{e}), \boldsymbol{e})$ such that $S(\boldsymbol{e}) = \boldsymbol{s}$ (if exists) can be found efficiently. A hash table can perform this search in constant time. The construction of this hash table is the pre–computation step of the algorithm. Then, the algorithm is elementary, given a received word $\boldsymbol{y}$ compute $\boldsymbol{H} \cdot \boldsymbol{y}$ and search in the hash table the pair $(S(\boldsymbol{e}), \boldsymbol{e})$ with the corresponding entry and return $\boldsymbol{y} - \boldsymbol{e}$.

Algorithm 3: Precomputation step of the Syndrome decoding

Input : A code $\mathscr{C}$ and its minimum distance d
Output: A dictionnary (hash table) H of pairs $(S(\boldsymbol{e}), \boldsymbol{e})$ for any $\boldsymbol{e}$ of weight less than $\frac{d-1}{2}$

```
Initialize a hash table H = ∅;
for w = 0 to ⌊(d − 1)/2⌋ do
    for e ∈ F_q^n of weight w do
        H ← H ∪ {(H · e, e)};
    end
end
return H;
```

Remark 18. The precomputation step can be performed without knowing the minimum distance of the code. It is only a bit more technical: as soon as two distinct vectors $\boldsymbol{e}, \boldsymbol{e}'$ with the same weight w have the same syndrome, this means that w is larger than $\frac{d-1}{2}$. Then remove from the hash table any entry $(S(\boldsymbol{e}), \boldsymbol{e})$ such that $\mathrm{w_H}(\boldsymbol{e}) = w$. Details are left to the reader.

Algorithm 4: Syndrome decoding

Input : A code $\mathscr{C}$, a hash table H of syndroms obtained by pre–computation, a vector $\boldsymbol{y} \in \mathbb{F}_q^n$
Output: A vector $\boldsymbol{c} \in \mathscr{C}$ at distance less than $\frac{d-1}{2}$ from $\boldsymbol{y}$ if exists. "?" if not.

```
s ← H · y;
if No pair (S(e), e) ∈ H satisfies S(e) = s then
    return "?";
else
    Let (S(e), e) ∈ H be such that S(e) = e;
    return y − e;
end
```

Remark 19. Actually it is easy to adapt this algorithm in order to be able to correct any error pattern of weight less than or equal to t for a fixed bound $t \leqslant \frac{d-1}{2}$.

The syndrome decoding solves the unambiguous problem. Its time complexity is polynomial since the computation of the syndrome is that of a matrix $\times$ vector multiplication which is $O(n(n-k))$ operations in $\mathbb{F}_q$ for the considered matrix and vectors. Moreover, the cost of searching in H is constant if H is a hash table.

Unfortunately, this decoding algorithm cannot be practically used to correct many errors since, its major drawback is its space complexity which is the size of the hash table:

$$|H| = \sum_{i=0}^{t} \left| \left\{ \boldsymbol{e} \in \mathbb{F}_q^n \mid \mathrm{w_H}(\boldsymbol{e}) = i \right\} \right|,$$

where $t = \frac{d-1}{2}$ (or less, see Remark 19). This size is

$$|H| = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i,$$

which is exponential in t. Therefore, the pre–computation step is exponential in time. Next, the whole algorithm (pre–computation and computation) has an exponential space complexity.

Chapter 3

What is doable? What is not? Shannon Theorem

In this chapter we address the question on the limits of error correction. A natural question, is :

> *Given a noisy channel, is optimal error correction possible? And if it does, is it possible with a nonzero information rate?*

Shannon Theorem addresses this question.
The main references for this chapter are other lecture notes: [Rud], [Gur10] and [Zém13].

Notation 3.1. In what follows, the binary symmetric channel and the q–ary symmetric channel of parameter p are respectively denoted by BSC(p) and qSC(p).

3.1 Prerequisites on probability theory

The following results are useful in what follows.

Theorem 3.1 (Markov inequality)**.** *Let X be a non negative random variable, then for all $a > 0$*

$$\mathbb{P}\Big(X \geqslant a\Big) \leqslant \frac{\mathbb{E}(X)}{a}.$$

Proof. Let $\mathbf{1}_{X \geqslant a}$ be the random variable satisfying

$$\mathbf{1}_{X \geqslant a} = \begin{cases} 1 & \text{if} \quad X \geqslant a, \\ 0 & \text{else.} \end{cases}$$

We have $\mathbb{E}(\mathbf{1}_{X \geqslant a}) = \mathbb{P}(X \geqslant a)$. Moreover, $a\mathbf{1}_{X \geqslant a} \leqslant X$. By applying the expected value to the last inequality we get

$$a\mathbb{P}\big(X \geqslant a\big) \leqslant \mathbb{E}(X).$$

□

Theorem 3.2 (Tchebychev inequality)**.** *Let X be a real random variable, then for all $a > 0$*

$$\mathbb{P}\big(|X - \mathbb{E}(X)| \geqslant a\big) \leqslant \frac{Var(X)}{a^2}.$$

Proof. Apply Markov inequality to the random variable $(X - \mathbb{E}(X))^2$. □

Application to the binary symmetric channel

Proposition 3.3. *Let $\boldsymbol{e} \in \mathbb{F}_q^n$ be an error vector produced by the q–ary symmetric channel, i.e $\boldsymbol{e} = (e_1, \ldots, e_n)$ where $e_1, \ldots, e_n$ are pairwise independent Bernouilli random variables. We have,*

$$\mathbb{E}(w_H(\boldsymbol{e})) = pn \quad \text{and} \quad Var(w_H(\boldsymbol{e})) = np(1-p).$$

Proof. For all i, $\mathbb{E}(e_i) = p$ and $\mathrm{Var}(e_i) = p(1-p)$. The expected value is obtained by summing up the n expected values. Since the variables are independent, summing up the variances provide the variance. □

Corollary 3.4. *Let $\boldsymbol{e} \in \mathbb{F}_q^n$ be an error vector produced by the q–ary symmetric channel, then, for all $\varepsilon > 0$,*

$$\mathbb{P}\Big(w_H(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant \frac{p(1-p)}{\varepsilon^2 n}.$$

Proof. We have,

$$\begin{aligned}\mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) &\leqslant \mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) + \mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \leqslant (p-\varepsilon)n\Big) \\ &\leqslant \mathbb{P}\Big(|\mathrm{w_H}(\boldsymbol{e}) - pn| \geqslant \varepsilon n\Big).\end{aligned}$$

Then, thanks to Tchebychev inequality, we get,

$$\mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant \frac{p(1-p)}{\varepsilon^2 n}.$$

□

This upper bound can actually be improved by:

Theorem 3.5 (Chernoff bound)**.** *Let $\boldsymbol{e} \in \mathbb{F}_q^n$ be an error vector produced by the q–ary symmetric channel. Then, for all $\varepsilon > 0$,*

$$\mathbb{P}\Big(w_H(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant e^{-\frac{pn\varepsilon^2}{3}}.$$

Proof. See Appendix A.1. □

3.2 Decoders

Remind that decoders have been introduced in Chapter 2. See Definition 2.1.

Caution. In the present chapter, when we speak about the existence of a decoder, **we never assert that the evaluation of this map is easy to compute**: the algorithm used by the decoder may have a huge complexity which would make it unsuitable for a practical use. In summary, in this chapter, we discuss the theoretical feasibility but no practical realization.

3.2.1 The failure probability

Definition 3.6. Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $\mathcal{D}$ be a decoder for $\mathscr{C}$, the *failure probability* of $\mathcal{D}$ is defined as

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}) \stackrel{\text{def}}{=} \mathbb{P}_{\boldsymbol{e}\sim\text{qSC}(p),\ \boldsymbol{c}\sim\mathbb{U}(\mathscr{C})} \left(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c}\right),$$

where "$\boldsymbol{c} \sim \mathbb{U}(\mathscr{C})$" means that $\boldsymbol{c}$ is a uniformly random element of $\mathscr{C}$ and the random variables $\boldsymbol{c}$ and $\boldsymbol{e}$ are independent.

The failure probability of a decoder quantifies its efficiency. The smaller the failure probability, the better the pair $(\mathscr{C}, \mathcal{D})$.

3.2.2 The maximum likelihood decoder for the q–ary symmetric channel

The maximum likelihood decoder has been introduced in § 2.2.3. Moreover, it is observed in § 2.2.3 that this decoder is nothing but the one who returns the closest codeword to the entry (if unique) with respect to the Hamming distance. Let us introduce the following notation for this decoder:

Notation 3.2. The maximum likelihood decoder is denoted as $\mathcal{D}^{\text{ML}}$. That is,

$$\forall \boldsymbol{y} \in \mathbb{F}_2^n, \mathcal{D}^{\text{ML}}(\boldsymbol{y}) \stackrel{\text{def}}{=} \begin{cases} \boldsymbol{c} & \text{if } \boldsymbol{c} \text{ is the unique element of } \mathscr{C} \\ & \quad \text{satisfying } \mathrm{d_H}(\boldsymbol{c}, \boldsymbol{y}) = \min_{\boldsymbol{u}\in\mathscr{C}}\{\mathrm{d_H}(\boldsymbol{u}, \boldsymbol{y})\}; \\ \{?\} & \text{else.} \end{cases}$$

For a linear code $\mathscr{C}$ together with the maximum likelihood decoder, the expression of the failure probability is rather simple.

Lemma 3.7. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a linear code, then,*

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}^{\text{ML}}) = \mathbb{P}_{\boldsymbol{e}\sim qSC(p)}(\mathcal{D}^{\text{ML}}(\boldsymbol{e}) \neq 0), \tag{3.1}$$

or equivalently,

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}^{\text{ML}}) = \mathbb{P}_{\boldsymbol{e}\sim qSC(p)} \left(\exists \boldsymbol{u} \in \mathbf{B}_H(\boldsymbol{e}, w_H(\boldsymbol{e})) \cap \mathscr{C} \setminus \{0\}\right). \tag{3.2}$$

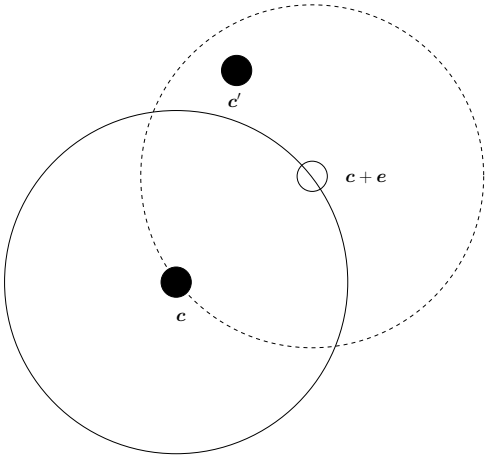


Figure 3.1: Situation of decoding failure for the maximum likelihood decoder

Proof. Let $\boldsymbol{c}$ be a uniformly randomly chosen element of $\mathscr{C}$ and $\boldsymbol{e}$ be an error vector given by the binary symmetric channel. The maximum likelihood decoder will fail if there exists another word in $\mathscr{C}$ at least as close as $\boldsymbol{c}$ to $\boldsymbol{c}+\boldsymbol{e}$ or equivalently if $\mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}+\boldsymbol{e}, \mathrm{w}_{\mathrm{H}}(\boldsymbol{e})\right) \cap \mathscr{C}$ contains a word distinct from $\boldsymbol{c}$. This situation is represented by Figure 3.1. Therefore, for a uniformly random $\boldsymbol{c} \in \mathscr{C}$,

$$\mathbb{P}_{\boldsymbol{e}\sim \mathrm{qSC}(p)}(\mathcal{D}^{\mathrm{ML}}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c}) = \mathbb{P}(\exists \boldsymbol{c}' \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}+\boldsymbol{e}, \mathrm{w}_{\mathrm{H}}(\boldsymbol{e})\right) \cap \mathscr{C} \setminus \{\boldsymbol{c}\}).$$

By linearity and after applying a translation by $-\boldsymbol{c}$, the above probability is nothing but

$$\mathbb{P}(\exists \boldsymbol{u} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, \mathrm{w}_{\mathrm{H}}(\boldsymbol{e})\right) \cap \mathscr{C} \setminus \{0\}).$$

This proves (3.2) and (3.1) is nothing but a reformulation of (3.2). □

3.3 Shannon Theorem

Let us start with a question already raised in the introduction.

Question 3.1. Does there exist a family of pairs $(\mathscr{C}_i, \mathcal{D}_i)_{i\in\mathbb{N}}$, where $\mathscr{C}_i \in \mathbb{F}_2^{n_i}$ is a code, $\mathcal{D}_i$ is a decoder for $\mathscr{C}_i$ and such that

$$\lim_{i\to+\infty} \mathbb{P}_{\mathrm{fail}}(\mathscr{C}_i, \mathcal{D}_i) = 0?$$

Roughly speaking: "is it possible to correct almost all errors arising from the channel?"

3.3.1 First idea: use the repetition code

Consider the following sequence $\left(\mathscr{C}_n, \mathcal{D}^{\mathrm{ML}}\right)_n$ where $\mathscr{C}_n$ is the $[n,1,n]_2$ repetition code. Notice that $\mathcal{D}^{\mathrm{ML}}$ here can be efficiently computed since in this situation it is noting but a majority

voting process:

$$\mathcal{D}^{\mathrm{ML}}(\boldsymbol{y}) = \begin{cases} (0 \ \cdots \ 0) & \text{if there is a majority of 0's in } \boldsymbol{y} \\ (1 \ \cdots \ 1) & \text{if there is a majority of 1's in } \boldsymbol{y} \\ ? & \text{if there is the same number of 0's and 1's in } \boldsymbol{y} \end{cases}$$

The probability failure of $\mathcal{D}^{\mathrm{ML}}$ is

$$\mathbb{P}_{\mathrm{fail}}(\mathscr{C}_n, \mathcal{D}_n) = \mathbb{P}_{\boldsymbol{e}\sim \mathrm{qSC}(p)}\left(\mathrm{w_H}(\boldsymbol{e}) \geqslant \frac{n}{2}\right). \tag{3.3}$$

Indeed the probability of a wrong decoding is nothing but the probability that a majority of bits are corrupted or equivalently the probability that the number of errors exceeds $\frac{n}{2}$ (i.e. $\mathrm{w_H}(\boldsymbol{e}) \geqslant \frac{n}{2}$).

Proposition 3.8. *For $p < 1/2$, $(\mathscr{C}_n, \mathcal{D}_n)_n$ described as above we have*

$$\lim_{n\to+\infty} \mathbb{P}_{\mathrm{fail}}(\mathscr{C}_n, \mathcal{D}_n) = 0.$$

Proof. Let $0 < \varepsilon < p$ such that $p + \varepsilon \leqslant 1/2$.

$$\begin{aligned} \mathbb{P}_{\mathrm{fail}}(\mathscr{C}_n, \mathcal{D}_n) &= \mathbb{P}_{\boldsymbol{e}\sim \mathrm{qSC}(p)}\left(\mathrm{w_H}(\boldsymbol{e}) \geqslant \frac{n}{2}\right) \\ &\leqslant \mathbb{P}_{\boldsymbol{e}}\big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\big). \end{aligned}$$

Thanks to Corollary 3.4, we get

$$\mathbb{P}_{\mathrm{fail}}(\mathscr{C}_n, \mathcal{D}_n) \leqslant \frac{p(1-p)}{\varepsilon^2 n},$$

which tends to 0 when n tends to infinity. □

Here we proved that, for a sufficiently large n, the repetition code together with the majority voting decoding permits a communication where almost all errors are corrected. Unfortunately, this approach has a huge drawback. Indeed, the information rate of the repetition code of length n is $\frac{1}{n}$. Hence, when n tends to infinity, the information rate of the code tends to 0. This motivates a refinement of Question 3.1.

Question 3.2. Does there exist a family of pairs $(\mathscr{C}_i, \mathcal{D}_i)_{i\in\mathbb{N}}$, where $\mathscr{C}_i \in \mathbb{F}_2^{n_i}$ is a linear code, $\mathcal{D}_i$ is a decoding algorithm for $\mathscr{C}_i$ and such that

$$\lim_{i\to+\infty} \mathbb{P}_{\mathrm{fail}}(\mathscr{C}_i, \mathcal{D}_i) = 0?$$

and the sequence of rates $R_i \overset{\mathrm{def}}{=} \frac{\dim \mathscr{C}_i}{n_i}$ is bounded below by a nonzero constant?

Shannon Theorem answers positively to this question. To state it, we first need to introduce the notion of entropy.

3.3.2 The entropy function

In information theory, the entropy of a channel is a function quantifying its uncertainty. For the binary symmetric channel, the entropy function is called the *binary entropy* and defined as follows

$$H_2(p) = \begin{cases} -p\log_2 p - (1-p)\log_2(1-p) & \text{if} \quad 0 < p < 1; \\ 0 & \text{if} \quad p = 0 \quad \text{or} \quad p = 1 \end{cases}$$

Basically, the higher the entropy, the higher the incertitude of the channel. Let us list a few remarks on this function.

- The entropy function is continuous and positive.
- The function has maximum for $p = \frac{1}{2}$. Clearly, if the error probability is $\frac{1}{2}$, we reach the maximal uncertainty.
- For all $p \in [0,1]$, we have $H_2(p) = H_2(1-p)$. This symmetry is explained by the fact that both probabilities yield the same uncertainty since one can flip the value of the outputted bits and then switch from a qSC(p) to a qSC$(1-p)$.

For the q–ary symmetric channel, one can define the q–ary entropy function as

$$H_q(p) = \begin{cases} p\log_q(q-1) - p\log_q p - (1-p)\log_q(1-p) & \text{if} \quad 0 < p < 1; \\ 0 & \text{if} \quad p = 0 \quad \text{or} \quad 1. \end{cases}$$

The q–ary entropy function reaches its maximum for $p = 1 - \frac{1}{q}$. Figure 3.2 represents the graph of the entropy function for $q = 2, 3, 5, 121$.

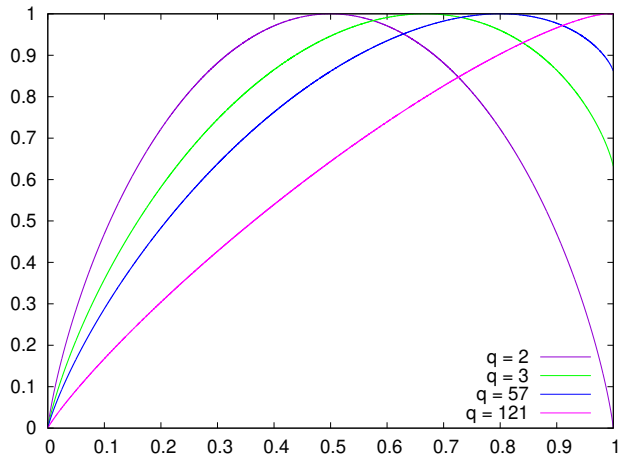


Figure 3.2: Graphs of entropy functions

3.3.3 Shannon Theorem

Theorem 3.9 (Shannon Theorem)**.** *For all* $0 < p < 1 - \frac{1}{q}$ *and all* $0 < \varepsilon < 1 - \frac{1}{q} - p$*, the following statements hold.*

1. *There exists* $\delta > 0$ *such that, for any large enough* n*, there exists a pair* $(\mathscr{C}, \mathcal{D})$ *where* $\mathscr{C}$ *is a code of length* n *and rate* $R = 1 - H_q(p) - \varepsilon$*,* $\mathcal{D}$ *is a decoder, and*

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}) < q^{-\delta n}.$$

2. *For all large enough* n *and all pair* $(\mathscr{C}, \mathcal{D})$*, where* $\mathscr{C}$ *has length* n *and rate* $R = 1 - H_q(p) + \varepsilon$ *and* $\mathcal{D}$ *is a decoder, then*

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}) \geqslant \frac{1}{2}.$$

Remark 20. Actually, the statement holds true even for non linear codes. We provide a proof of Theorem 3.9(1) only for linear codes. See [Rud] for a proof in the nonlinear case. For (2), our proof holds even for non linear codes.

Basically, this theorem points out the existence of a threshold called *channel capacity* and equal to $1 - H_q(p)$ in case of the qSC(p) channel. Moreover, the statement asserts that for communication rates below this capacity, then there is a suitable choice of a pair $(\mathscr{C}, \mathcal{D})$ providing an almost perfect communication. On the other hand, if you wish to communicate with rates above this threshold, then every choice of a pair $(\mathscr{C}, \mathcal{D})$ will provide a very bad communication.

Caution. Let us emphasize again that **Shannon Theorem is purely theoretic**. The first part of the statement asserts the existence of a pair $(\mathscr{C}, \mathcal{D})$ but is absolutely non constructive: it does not provide any method to compute or construct such a pair. Moreover, the decoder $\mathcal{D}$ is not necessarily an efficient decoding algorithm and can have a prohibitive complexity and be unreasonably time and space–consuming.

It should be noticed that this theorem has been proved in 1949, while practical and efficient realizations of codes with efficient decoding with rates close to the channel capacity and with very low error rates (after decoding) appeared only during the 90's, with LDPC codes and turbo codes.

To prove Shannon Theorem, some prerequisites are necessary and summarized in the forthcoming Sections 3.3.4, 3.1 and 3.3.5.

3.3.4 The asymptotic volume of Hamming balls

Recall that for all $\boldsymbol{x} \in \mathbb{F}_q^n$ and $0 \leqslant r \leqslant n$, the Hamming ball of radius r and center $\boldsymbol{x}$ is denoted by $\mathbf{B}_{\mathrm{H}}(\boldsymbol{x}, r)$.

Lemma 3.10 (Volume of a Hamming ball)**.** *Let $n \in \mathbb{N}$, $\boldsymbol{x} \in \mathbb{F}_q^n$ and $0 \leqslant r \leqslant n$. Then,*

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

Proof. Exercise. □

Notation 3.3. Notice that this volume does not depend on the center $\boldsymbol{x}$ but only on r and n, hence from now on, we denote it by $\mathrm{Vol}_q(r, n)$.

Lemma 3.11. *Let $n \in \mathbb{N}$ and $0 \leqslant p \leqslant 1 - \frac{1}{q}$ such that $pn \in \mathbb{N}$. Then,*

(1) $Vol_q(pn, n) \leqslant q^{nH_q(p)}$.

(2) $\forall \varepsilon > 0$, there exists $N > 0$ such that,

$$\forall n > N, \;\; Vol_q(pn, n) \geqslant q^{n(H_q(p)-\varepsilon)}.$$

Proof. See Appendix A.2. □

3.3.5 Random codes

A *random $[n, k]_q$ code* is a uniformly random element of the set of linear codes of length n, dimension k over $\mathbb{F}_q$. The following statement provides another definition which useful in what follows.

Lemma 3.12. *Let $\boldsymbol{G}$ be a uniformly random $k \times n$ matrix of rank k. Then, $Im(\boldsymbol{G}) \stackrel{def}{=} \{\boldsymbol{mG} \mid \boldsymbol{m} \in \mathbb{F}_q^k\}$ is a random code.*

Proof. We only have to prove that for all pairs $\mathscr{C}, \mathscr{C}'$ of codes of length n and dimension k,

$$\mathbb{P}(\mathrm{Im}(\boldsymbol{G}) = \mathscr{C}) = \mathbb{P}(\mathrm{Im}(\boldsymbol{G}) = \mathscr{C}').$$

A classical argument of linear algebra asserts that there exists a $n \times n$ invertible matrix $\boldsymbol{M}$ such that $\mathscr{C}\boldsymbol{M} = \mathscr{C}'$, where $\mathscr{C}\boldsymbol{M} \stackrel{\text{def}}{=} \{\boldsymbol{cM} \mid \boldsymbol{c} \in \mathscr{C}\}$. Therefore, $\mathrm{Im}(\boldsymbol{G}) = \mathscr{C}$ if and only if $\mathrm{Im}(\boldsymbol{GM}) = \mathscr{C}\boldsymbol{M} = \mathscr{C}'$. Hence

$$\mathbb{P}(\mathrm{Im}(\boldsymbol{G}) = \mathscr{C}) = \mathbb{P}(\mathrm{Im}(\boldsymbol{GM}) = \mathscr{C}'). \tag{3.4}$$

In addition, $\boldsymbol{GM}$ is a uniformly random full-rank matrix since the map $\boldsymbol{G} \mapsto \boldsymbol{GM}$ is a bijection of the set of full-rank $k \times n$ matrices onto itself. Therefore

$$\mathbb{P}(\mathrm{Im}(\boldsymbol{GM}) = \mathscr{C}) = \mathbb{P}(\mathrm{Im}(\boldsymbol{G}) = \mathscr{C}). \tag{3.5}$$

Combining (3.4) and (3.5), we get the result. □

Lemma 3.13. *Let $\boldsymbol{G}$ be a random full-rank $k \times n$ matrix. For all $\boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\}$ the random variable $\boldsymbol{mG}$ is uniformly distributed in $\mathbb{F}_q^n \setminus \{0\}$.*

Roughly speaking the above statement asserts that every nonzero element of a random $[n,k]_q$ code is a uniformly random element of $\mathbb{F}_q^n$.

Proof. Let $\boldsymbol{y}, \boldsymbol{y}'$ be two elements of $\mathbb{F}_q^n \setminus \{0\}$, we will prove that

$$\mathbb{P}(\boldsymbol{mG} = \boldsymbol{y}) = \mathbb{P}(\boldsymbol{mG} = \boldsymbol{y}').$$

Indeed, there exists an invertible $n \times n$ matrix $\boldsymbol{M}$ such that $\boldsymbol{yM} = \boldsymbol{y}'$. Therefore,

$$\mathbb{P}(\boldsymbol{xG} = \boldsymbol{y}) = \mathbb{P}(\boldsymbol{xGM} = \boldsymbol{y}').$$

Moreover, since $\boldsymbol{A} \mapsto \boldsymbol{AM}$ is a bijection from the set of full rank $k \times n$ matrices onto itself, $\boldsymbol{GM}$ is a uniformly random full rank $k \times n$ matrix and hence

$$\mathbb{P}(\boldsymbol{xGM} = \boldsymbol{y}') = \mathbb{P}(\boldsymbol{xG} = \boldsymbol{y}').$$

Combining the previous inequalities yields the result. □

3.3.6 Proof of Shannon Theorem

Proof of Theorem 3.9(1)

Set $k = \lfloor (1 - H_q(p) - \varepsilon)n \rfloor$. Let $\mathscr{C}$ be an $[n,k]_q$ code and $\boldsymbol{G}$ be a generator matrix for $\mathscr{C}$, i.e. $\mathscr{C} = \mathrm{Im}(\boldsymbol{G})$. Remind, that, from Lemma 3.7,

$$\mathbb{P}_{\mathrm{fail}}(\mathrm{Im}(\boldsymbol{G}), \mathcal{D}^{\mathrm{ML}}) = \mathbb{P}_{\boldsymbol{e} \sim \mathrm{qSC}(p)}\Big(\exists \boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\},\ \boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, \mathrm{w}_{\mathrm{H}}(\boldsymbol{e})\right)\Big).$$

Denote by $\mathcal{E}(\boldsymbol{G}, \boldsymbol{e})$ the event:

$$\mathcal{E}(\boldsymbol{G}, \boldsymbol{e}) \stackrel{\mathrm{def}}{=} \left\{\exists \boldsymbol{m} \in (\mathbb{F}_q^k \setminus \{0\}),\ \boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, \mathrm{w}_{\mathrm{H}}(\boldsymbol{e})\right)\right\}.$$

Let $\gamma > 0$, then,

$$\begin{aligned}\mathbb{P}_{\mathrm{fail}}(\mathrm{Im}(\boldsymbol{G}), \mathcal{D}^{\mathrm{ML}}) = {} & \mathbb{P}_{\boldsymbol{e}}\Big(\mathcal{E}(\boldsymbol{G}, \boldsymbol{e}) \;\Big|\; \mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big)\mathbb{P}_{\boldsymbol{e}}\Big(\mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big) \\ & + \mathbb{P}_{\boldsymbol{e}}\Big(\mathcal{E}(\boldsymbol{G}, \boldsymbol{e}) \;\Big|\; \mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) > (p+\gamma)n\Big)\mathbb{P}_{\boldsymbol{e}}\Big(\mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) > (p+\gamma)n\Big).\end{aligned}$$

Since probabilities are always less than or equal to 1, we get

$$\mathbb{P}_{\mathrm{fail}}(\mathrm{Im}(\boldsymbol{G}), \mathcal{D}^{\mathrm{ML}}) \leqslant \mathbb{P}_{\boldsymbol{e}}\Big(\mathcal{E}(\boldsymbol{G}, \boldsymbol{e}) \;\Big|\; \mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big) + \mathbb{P}_{\boldsymbol{e}}\Big(\mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) > (p+\gamma)n\Big). \tag{3.6}$$

Thanks to Chernoff bound (Theorem 3.5),

$$\mathbb{P}_{\boldsymbol{e}}\Big(\mathrm{w}_{\mathrm{H}}(\boldsymbol{e}) > (p+\gamma)n\Big) \leqslant e^{-\frac{pn\gamma^2}{3}}. \tag{3.7}$$

There remains to get an upper bound on $\mathbb{P}_{\boldsymbol{e}}\Big(\mathcal{E}(\boldsymbol{G},\boldsymbol{e}) \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big)$. However, it is hopeless to get a uniform and sharp upper bound for all $\boldsymbol{G}$. Therefore, to prove that it is small for some matrices $\boldsymbol{G}$ (or equivalently for some codes), we will first prove that it is small *in average*. Thus, let $\mathcal{G}_{k,n}$ be the set of full rank $k \times n$ matrices and consider the mean:

$$\begin{aligned} M &\stackrel{\text{def}}{=} \frac{1}{|\mathcal{G}_{k,n}|} \sum_{\boldsymbol{G}\in\mathcal{G}_{k,n}} \mathbb{P}\Big(\exists \boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\},\ \boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, \mathrm{w_H}(\boldsymbol{e})\right) \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big) \\ &\leqslant \frac{1}{|\mathcal{G}_{k,n}|} \sum_{\boldsymbol{G}\in\mathcal{G}_{k,n}} \mathbb{P}\Big(\exists \boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\},\ \boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, (p+\gamma)n\right) \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big) \end{aligned}$$

By the union bound, we get

$$M \leqslant \frac{1}{|\mathcal{G}_{k,n}|} \sum_{\boldsymbol{G}\in\mathcal{G}_{k,n}} \sum_{\boldsymbol{m}\in\mathbb{F}_q^k\setminus\{0\}} \mathbb{P}\Big(\boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, (p+\gamma)n\right) \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big)$$

Therefore,

$$M \leqslant \frac{1}{|\mathcal{G}_{k,n}|} \sum_{\boldsymbol{G}} \sum_{\boldsymbol{m}} \sum_{\boldsymbol{x}:\mathrm{w_H}(\boldsymbol{x})\leqslant(p+\gamma)n} \mathbb{P}\Big(\boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{x}, (p+\gamma)n\right)\Big)\mathbb{P}\Big(\boldsymbol{e}=\boldsymbol{x} \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big).$$

If $\boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\}$ and $\boldsymbol{G} \in \mathcal{G}_{k,n}$ are arbitrary, then, from Lemma 3.13, the word $\boldsymbol{mG}$ is uniformly random in $\mathbb{F}_q^n \setminus \{0\}$. Therefore,

$$\mathbb{P}\Big(\boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{x}, (p+\gamma)n\right)\Big) = \frac{\mathrm{Vol}((p+\gamma)n, n)}{q^n} \leqslant q^{n(H_q(p+\gamma)-1)},$$

where the last inequality is a direct consequence of Lemma 3.11(1). Consequently,

$$\begin{aligned} M &\leqslant \frac{1}{|\mathcal{G}_{k,n}|} \sum_{\boldsymbol{G}} \sum_{\boldsymbol{m}} \sum_{\boldsymbol{x}:\mathrm{w_H}(\boldsymbol{x})\leqslant(p+\gamma)n} q^{n(H_q(p+\gamma)-1)} \mathbb{P}\Big(\boldsymbol{e}=\boldsymbol{x} \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big) \\ &\leqslant q^{n(H_q(p+\gamma)-1)} \underbrace{\left(\frac{1}{|\mathcal{G}_{k,n}|} \sum_{\boldsymbol{G}} \sum_{\boldsymbol{m}} 1\right)}_{=q^k-1} \cdot \underbrace{\left(\sum_{\boldsymbol{x}:\mathrm{w_H}(\boldsymbol{x})\leqslant(p+\gamma)n} \mathbb{P}\Big(\boldsymbol{e}=\boldsymbol{x} \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big)\right)}_{=1} \end{aligned}$$

Finally, since, $k \leqslant n(1-H_q(p)-\varepsilon)$, we conclude that

$$M \leqslant q^{n(H_q(p+\gamma)-H_q(p)-\varepsilon)}. \tag{3.8}$$

Since the mean is taken over all the full-rank $n \times k$ matrices, there exists a full rank $n \times k$ matrix $\boldsymbol{G}$ such that:

$$\mathbb{P}\Big(\exists \boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\},\ \boldsymbol{mG} \in \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{e}, \mathrm{w_H}(\boldsymbol{e})\right) \;\Big|\; \mathrm{w_H}(\boldsymbol{e}) \leqslant (p+\gamma)n\Big) \leqslant q^{n(H_q(p+\gamma)-H_q(p)-\varepsilon)}. \tag{3.9}$$

Putting together (3.9), (3.7) and (3.6), we get

$$\begin{aligned}\mathbb{P}_{\text{fail}}(\text{Im}(\boldsymbol{G}), \mathcal{D}^{\text{ML}}) &\leqslant q^{n(H_q(p+\gamma)-H_q(p)-\varepsilon)} + e^{-\frac{pn\gamma^2}{3}} \\ &\leqslant q^{n(H_q(p+\gamma)-H_q(p)-\varepsilon)} + q^{-\frac{pn\gamma^2}{3\log q}}.\end{aligned} \tag{3.10}$$

Since the function H_q is continuous on $[0, 1-\frac{1}{q}]$, for γ small enough, $H_q(p+\gamma)-H_q(p)-\varepsilon < 0$. Therefore, for γ small enough, the exponents of both terms of the right hand side of (3.10) are negative. Thus, for some positive constant δ and for n large enough, we get

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}^{\text{ML}}) \leqslant q^{-\delta n}.$$

This concludes the proof.

Remark 21. Actually, one could have been more precise. The above proof asserts that "almost all codes" together with the maximum likelihood decoder has an exponentially small failure probability. Indeed, for a random code $\mathscr{C}$, Equations (3.9), (3.7) and (3.6) can be reformulated as

$$\mathbb{E}\left(\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}^{\text{ML}})\right) \leqslant q^{-\delta n}$$

for a positive constant δ and n large enough. Indeed, let $\mathscr{C}$ be a random code, then, using Markov inequality,

$$\mathbb{P}\left(\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}^{\text{ML}}) \geqslant q^{-\frac{\delta n}{2}}\right) \leqslant \frac{\mathbb{E}(\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}^{\text{ML}}))}{q^{-\frac{\delta n}{2}}} = q^{-\frac{\delta n}{2}}.$$

Therefore, for a large enough n, with a probability close to 1, the failure probability of a random code with the maximum likelihood decoder is $\leqslant q^{-\frac{\delta n}{2}}$.

Proof of Theorem 3.9(2)

Assume the result is wrong and hence assume that for all N positive, there exists a pair $(\mathscr{C}_n, \mathcal{D}_n)$ of length $n \geqslant N$, of rate $1 - H_q(p) + \varepsilon$ and such that $\mathbb{P}_{\text{fail}}(\mathscr{C}_n, \mathcal{D}_n) < \frac{1}{2}$. For convenience sake we omit the indexes "n" and refer to the pair $(\mathscr{C}, \mathcal{D})$.

The general idea of the proof works as follows, consider for all $\boldsymbol{c} \in \mathscr{C}$ the set of words

$$\mathcal{D}^{-1}(\boldsymbol{c}) \stackrel{\text{def}}{=} \left\{\boldsymbol{x} \in \mathbb{F}_q^n \mid \mathcal{D}(\boldsymbol{x}) = \boldsymbol{c}\right\}.$$

The subsets $\mathcal{D}^{-1}(\boldsymbol{c})$ are pairwise disjoint, and the assumption "$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}) < \frac{1}{2}$" entails that these sets are "large". In addition, the assumption "$\mathscr{C}$ has rate $1 - H_q(p) + \varepsilon$" entails that there are *too many* such sets. Indeed, we will prove that the union of these sets has a volume larger than q^n which is a contradiction.

For this sake we will give a lower bound for the average volume of $\mathcal{D}^{-1}(\boldsymbol{c})$.

Proposition 3.14. *Let $\boldsymbol{c}$ be a random variable uniformly distributed over $\mathscr{C}$. Then,*

$$\mathbb{E}(|\mathcal{D}^{-1}(\boldsymbol{c})|) \geqslant \frac{1}{4} q^{-nH_q(p)}.$$

Proof. Let $\boldsymbol{c}_0 \in \mathscr{C}$ and let $S_{\boldsymbol{c}_0}$ be the set

$$S_{\boldsymbol{c}_0} \stackrel{\text{def}}{=} \left\{ \boldsymbol{x} \in \mathbb{F}_q^n \;\middle|\; |\mathrm{d_H}(\boldsymbol{c}_0, \boldsymbol{x}) - pn| < \sqrt{n} \right\}.$$

The set $S_{\boldsymbol{c}_0,\gamma}$ is the "shell" $\mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}_0, (pn + \sqrt{n}\right) \setminus \mathbf{B}_{\mathrm{H}}\left(\boldsymbol{c}_0, (pn - \sqrt{n}\right)$ as illustrated in Figure 3.3.

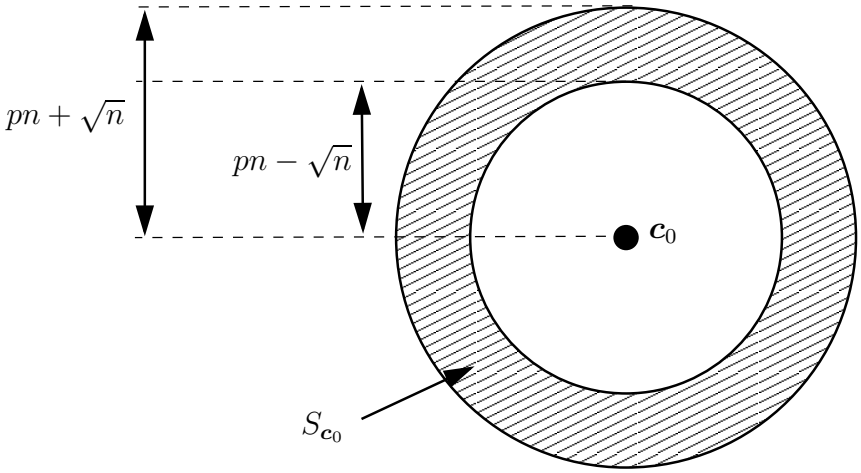


Figure 3.3: The shell $S_{\boldsymbol{c}_0}$

Consider the following probability,

$$\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0 + \boldsymbol{e} \in S_{\boldsymbol{c}_0} \cap \mathcal{D}^{-1}(\boldsymbol{c}_0)) = \sum_{\boldsymbol{x} \in S_{\boldsymbol{c}_0} \cap \mathcal{D}^{-1}(\boldsymbol{c}_0)} \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0 + \boldsymbol{e} = \boldsymbol{x}) \tag{3.11}$$

$$\leqslant |S_{\boldsymbol{c}_0} \cap \mathcal{D}^{-1}(\boldsymbol{c}_0)| \max_{\boldsymbol{x} \in S_{\boldsymbol{c}_0}} \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0 + \boldsymbol{e} = \boldsymbol{x}). \tag{3.12}$$

Next, notice that for all $\boldsymbol{x} \in S_{\boldsymbol{c}_0}$,

$$\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0 + \boldsymbol{e} = \boldsymbol{x}) = p^{\mathrm{d_H}(\boldsymbol{x},\boldsymbol{c}_0)}(q-1)^{-\mathrm{d_H}(\boldsymbol{x},\boldsymbol{c}_0)}(1-p)^{n-\mathrm{d_H}(\boldsymbol{x},\boldsymbol{c}_0)}.$$

Then, consider the function $f : x \mapsto p^x(q-1)^x(1-p)^{n-x}$. This function is positive on $]0, n[$ and its derivative satisfies $f'(x) = \log\left(\frac{p(q-1)}{1-p}\right) f(x)$. Therefore, if $\frac{p(q-1)}{1-p} > 1$, then f is increasing else it is decreasing. Thus,

$$\max_{\boldsymbol{x} \in S_{\boldsymbol{c}_0}} \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0 + \boldsymbol{e} = \boldsymbol{x}) = \begin{cases} p^{pn+\sqrt{n}}(q-1)^{pn+\sqrt{n}}(1-p)^{1-(pn+\sqrt{n})} & \text{if} \quad \frac{p(q-1)}{1-p} > 1 \\ p^{pn-\sqrt{n}}(q-1)^{pn-\sqrt{n}}(1-p)^{1-(pn-\sqrt{n})} & \text{else.} \end{cases}$$

This entails,

$$\max_{\boldsymbol{x} \in S_{\boldsymbol{c}_0}} \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0 + \boldsymbol{e} = \boldsymbol{x}) = \begin{cases} q^{-nH_q(p)}\left(\frac{p(q-1)}{1-p}\right)^{-\sqrt{n}} & \text{if} \quad \frac{p(q-1)}{1-p} > 1 \\ q^{-nH_q(p)}\left(\frac{p(q-1)}{1-p}\right)^{\sqrt{n}} & \text{else,} \end{cases}$$

which leads to the upper bound

$$\max_{\boldsymbol{x}\in S_{\boldsymbol{c}_0}} \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}=\boldsymbol{x}) \leqslant q^{-nH_q(p)}. \tag{3.13}$$

Putting (3.12) and (3.13) together and using the obvious inclusion $\mathcal{D}^{-1}(\boldsymbol{c}_0)\cap S_{\boldsymbol{c}_0} \subseteq \mathcal{D}^{-1}(\boldsymbol{c}_0)$, we get

$$|\mathcal{D}^{-1}(\boldsymbol{c}_0)| \geqslant \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in S_{\boldsymbol{c}_0}\cap\mathcal{D}^{-1}(\boldsymbol{c}_0))\; q^{nH_q(p)} \tag{3.14}$$

and there remains to bound below $\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in S_{\boldsymbol{c}_0}\cap\mathcal{D}^{-1}(\boldsymbol{c}_0))$.

Remind that, from Proposition 3.3, the random variable $\mathrm{w_H}(\boldsymbol{e})$ has mean pn and variance $np(1-p)$. Thus, from Tchebychev inequality (Theorem 3.2),

$$\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\notin S_{\boldsymbol{c}_0}) = \mathbb{P}(|\mathrm{w_H}(\boldsymbol{e})-pn|\leqslant\sqrt{n}) \leqslant \frac{\mathrm{Var}(\mathrm{w_H}(\boldsymbol{e}))}{n} = p(1-p)\cdot$$

Moreover, since $p<1/2$ we have

$$\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\notin S_{\boldsymbol{c}_0}) < \frac{1}{4}\cdot \tag{3.15}$$

On the other hand,

$$\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in S_{\boldsymbol{c}_0}\cap\mathcal{D}^{-1}(\boldsymbol{c}_0)) = 1-\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in(\mathbb{F}_q^n\setminus S_{\boldsymbol{c}_0})\cup(\mathbb{F}_q^n\setminus\mathcal{D}^{-1}(\boldsymbol{c}_0)).$$

Then, by the union bound,

$$\begin{aligned}\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in S_{\boldsymbol{c}_0}\cap\mathcal{D}^{-1}(\boldsymbol{c}_0)) &\geqslant 1-\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\notin S_{\boldsymbol{c}_0})-\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\notin\mathcal{D}^{-1}(\boldsymbol{c}_0))\\ &\geqslant \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in\mathcal{D}^{-1}(\boldsymbol{c}_0))-\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\notin S_{\boldsymbol{c}_0}).\end{aligned}$$

Consequently, from (3.15),

$$\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in S_{\boldsymbol{c}_0}\cap\mathcal{D}^{-1}(\boldsymbol{c}_0)) \geqslant \mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}_0+\boldsymbol{e}\in\mathcal{D}^{-1}(\boldsymbol{c}_0))-\frac{1}{4}\cdot \tag{3.16}$$

Assume for now the following fact which is proved further (see Lemma 3.15). Let $\boldsymbol{c}$ be a random variable uniformly distributed over $\mathscr{C}$, which is independent from $\boldsymbol{e}$ then

$$\mathbb{P}_{\mathrm{fail}}(\mathscr{C},\mathcal{D}) = \mathbb{P}_{\boldsymbol{c},\boldsymbol{e}}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c}) = \mathbb{E}_{\boldsymbol{c}}(\mathbb{P}_{\boldsymbol{e}}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c})).$$

Consequently, (3.16) yields

$$\mathbb{E}_{\boldsymbol{c}}\left(\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}+\boldsymbol{e}\in S_{\boldsymbol{c}}\cap\mathcal{D}^{-1}(\boldsymbol{c}))\right) \geqslant 1-\mathbb{P}_{\mathrm{fail}}(\mathscr{C},\mathcal{D})-\frac{1}{4}\cdot$$

By assumption, $\mathbb{P}_{\mathrm{fail}}(\mathscr{C},\mathcal{D})\leqslant\frac{1}{2}$ and hence

$$\mathbb{E}_{\boldsymbol{c}}\left(\mathbb{P}_{\boldsymbol{e}}(\boldsymbol{c}+\boldsymbol{e}\in S_{\boldsymbol{c}}\cap\mathcal{D}^{-1}(\boldsymbol{c}))\right) \geqslant \frac{1}{4}.$$

Applying the expected value to (3.14) and using the above inequality, we get the result. □

In the previous proof, we assumed the following result, which we prove now.

Lemma 3.15. *Let $\boldsymbol{c}$ be a random variable uniformly distributed over $\mathscr{C}$ and $\boldsymbol{e} \sim qSC(p)$ such that $\boldsymbol{c}, \boldsymbol{e}$ are independent. Then*

$$\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}) = \mathbb{P}_{\boldsymbol{c},\boldsymbol{e}}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c}) = \mathbb{E}_{\boldsymbol{c}}(\mathbb{P}_{\boldsymbol{e}}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c})).$$

Proof. The first equality is the very definition of $\mathbb{P}_{\text{fail}}$. Then,

$$\begin{aligned}\mathbb{P}_{\boldsymbol{c},\boldsymbol{e}}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c}) &= \mathbb{E}_{\boldsymbol{c},\boldsymbol{e}}(\mathbf{1}_{\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c}}) \\ &= \sum_{\boldsymbol{c}_0\in\mathscr{C},\boldsymbol{e}_0\in\mathbb{F}_q^n} \mathbb{P}(\{\boldsymbol{c}=\boldsymbol{c}_0\}\cap\{\boldsymbol{e}=\boldsymbol{e}_0\})\mathbf{1}_{\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c}}.\end{aligned}$$

Since $\boldsymbol{c}, \boldsymbol{e}$ are independent, $\mathbb{P}(\{\boldsymbol{c}=\boldsymbol{c}_0\}\cap\{\boldsymbol{e}=\boldsymbol{e}_0\}) = \mathbb{P}(\boldsymbol{c}=\boldsymbol{c}_0)\mathbb{P}(\boldsymbol{e}=\boldsymbol{e}_0)$ and hence,

$$\begin{aligned}\mathbb{P}_{\boldsymbol{c},\boldsymbol{e}}(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c}) &= \sum_{\boldsymbol{c}_0\in\mathscr{C}} \mathbb{P}(\boldsymbol{c}=\boldsymbol{c}_0)\left(\sum_{\boldsymbol{e}_0\in\mathbb{F}_q^n} \mathbb{P}(\boldsymbol{e}=\boldsymbol{e}_0)\mathbf{1}_{\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c}}\right) \\ &= \mathbb{E}_{\boldsymbol{c}}\left(\mathbb{E}_{\boldsymbol{e}}\left(\mathbf{1}_{\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c}}\right)\right) \\ &= \mathbb{E}_{\boldsymbol{c}}\left(\mathbb{P}_{\boldsymbol{e}}\left(\mathcal{D}(\boldsymbol{c}+\boldsymbol{e})\neq\boldsymbol{c}\right)\right).\end{aligned}$$

□

Now we can conclude the proof of the second part of Shannon Theorem. Since the sets $\mathcal{D}^{-1}(\boldsymbol{c}_0)$ are pairwise disjoint when $\boldsymbol{c}_0$ varies over the elements of $\mathscr{C}$, we have

$$|\mathbb{F}_q^n| = q^n \geqslant \left|\bigcup_{\boldsymbol{c}_0\in\mathscr{C}} \mathcal{D}^{-1}(\boldsymbol{c}_0)\right| = \sum_{\boldsymbol{c}_0\in\mathscr{C}} |\mathcal{D}^{-1}(\boldsymbol{c}_0)|$$

and if $\boldsymbol{c}$ is a uniformly distributed random variable over $\mathscr{C}$, we have

$$\mathbb{E}(|\mathcal{D}^{-1}(\boldsymbol{c})|) = \sum_{\boldsymbol{c}_0\in\mathscr{C}} \mathbb{P}(\boldsymbol{c}=\boldsymbol{c}_0)|\mathcal{D}^{-1}(\boldsymbol{c}_0)| = \frac{1}{|\mathscr{C}|}\sum_{\boldsymbol{c}_0\in\mathscr{C}} |\mathcal{D}^{-1}(\boldsymbol{c}_0)|$$

Therefore,

$$q^n \geqslant |\mathscr{C}| \cdot \mathbb{E}_{\boldsymbol{c}}(|\mathcal{D}^{-1}(\boldsymbol{c})|).$$

By assumption, $|\mathscr{C}| = q^{n(1-H_q(p)+\varepsilon)}$. Then, thanks to Proposition 3.14, the above inequality yields

$$q^n \geqslant \frac{1}{4}q^{n(1+\varepsilon)}$$

which is a contradiction for n large enough.

Remark 22. Actually, the statement of Theorem 3.9 (2) could be improved as "for all pair $(\mathscr{C}, \mathcal{D})$, ..., then $\mathbb{P}_{\text{fail}}(\mathscr{C}, \mathcal{D}) \geqslant 1-\delta$ for all $\delta > 0$." To prove this improved version, replace the $\sqrt{n}$ by a $n^{3/4}$ in the proof of Proposition 3.14. Details are left to the reader.

Chapter 4

Bounds on codes

Problem We wish to produce a code $\mathscr{C} \subseteq \mathbb{F}_q^n$ with a high rate, and a high relative distance. That is with dimension and distance as close as possible from n. Unfortunately, these requirements contradict each other.

The point of the present chapter is to introduce several upper bounds on the minimum distance (resp. the dimension) of codes of fixed length and dimension (resp. minimum distance). Next we will focus on a *lower bound*: the famous Gilbert Varshamov bound. We should be careful that lower bounds have not the same status as upper bounds. An upper bound asserts that the parameters of *every code* lie below some bound while a lower bound asserts the existence of at *least one code* whose parameters exceed the bound.

4.1 Upper bounds

4.1.1 Singleton bound

The most elementary and probably the most famous upper bound on the parameters of codes is the Singleton bound.

Theorem 4.1 (Singleton bound)**.** *For every code $\mathscr{C}$ with parameters $[n,k,d]_q$, we have*

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

Proof. Let $\mathscr{C}$ be a code with parameters $[n,k,d]_q$. Let

$$\phi : \left\{ \begin{array}{ccc} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^{n-(d-1)} \\ (x_1, \ldots, x_n) & \longmapsto & (x_1, \ldots, x_{n-(d-1)}) \end{array} \right. .$$

We claim that the restriction $\phi_{|\mathscr{C}}$ of ϕ to $\mathscr{C}$ is injective. Indeed, if $\boldsymbol{x} \in \mathscr{C}$ satisfies $\phi(\boldsymbol{x}) = 0$, then $x_1 = \cdots = x_{n-(d-1)} = 0$ and hence $w_H(\boldsymbol{x}) < d$, which entails $\boldsymbol{x} = 0$.

Since $\phi_{|\mathscr{C}} : \mathscr{C} \to \mathbb{F}_q^{n-(d-1)}$ is injective, then

$$k = \dim(\mathscr{C}) = \dim \phi(\mathscr{C}) \leqslant n - d + 1,$$

which yields the result. □

Remark 23. An alternative and more explicit proof can be obtained by Gaussian elimination. Consider a code of length n and dimension k and consider a full rank generator matrix $\boldsymbol{G} \in \mathfrak{M}_{k,n}(\mathbb{F}_q)$ for $\mathscr{C}$. That is, the rows of $\boldsymbol{G}$ form a basis for $\mathscr{C}$. Note that performing row operations on $\boldsymbol{G}$ do not change the code and hence provide another generator matrix for the **same** code.

The, perform Gaussian elimination on $\boldsymbol{G}$ to put it in row echelon form. After elimination, the k–th row has weight $\leqslant n-k+1$ and is a nonzero codeword. This concludes the proof.

Remark 24. The above proof is suitable only for linear codes while the Singleton bound holds true for nonlinear codes. In the context of nonlinear codes it asserts that for all $\mathscr{C} \subset \mathbb{F}_q^n$, possibly nonlinear,

$$|\mathscr{C}| \leqslant q^{n-d+1}.$$

Definition 4.2. A code is said to be *Maximum Distance Separable* (MDS) if it reaches the Singleton bound.

Comments The Singleton bound is sharp for short codes. In particular, we will see in the next chapters, that for all $n \leqslant q+1$, there always exists an MDS code of length n. On the other hand, the existence of longer MDS codes is still partially open. It is actually conjectured that out of some degenerate cases there is no MDS code over $\mathbb{F}_q$ of length strictly higher than $q+1$.

Finally, for fixed q and high length compared to q, many other upper bounds are sharper than Singleton bound.

Asymptotic Singleton bound It can be useful to consider bounds asymptotically. This point of view is not that relevant for the Singleton bound since, as explained earlier, it is far from being sharp for long codes. Nevertheless, let us give an asymptotical bound which is actually elementary to obtain.

Lemma 4.3. *Let $(\mathscr{C}_r)_{r\in\mathbb{N}}$ be a sequence of codes of parameters $[n_r, k_r, d_r]_q$ over a fixed base field $\mathbb{F}_q$ such that n_r tends to infinity. Assume moreover that the following limits exist:*

$$R \overset{def}{=} \lim_{r\to+\infty} \frac{k_r}{n_r} \qquad \text{and} \qquad \delta \overset{def}{=} \lim_{r\to+\infty} \frac{d_r}{n_r}.$$

Then,

$$R + \delta \leqslant 1.$$

4.1.2 Hamming or sphere packing bound

As suggested this bound reposes on the notion of "sphere packing". It is actually a straightforward application of Lemma 1.4 asserting that the balls of radius $\lfloor \frac{d-1}{2} \rfloor$ are pairwise disjoint. For an $[n,k,d]_q$ code $\mathscr{C}$, we get q^k disjoint Hamming balls of radius $\lfloor \frac{d-1}{2} \rfloor$. The volume of their union should be less than the total volume q^n which yields:

Theorem 4.4 (The Hamming or Sphere Packing bound)**.** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be an $[n,k,d]_q$ code. Then,*

$$q^k \, Vol_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right) \leqslant q^n.$$

(See Notation 3.3 for the definition of $Vol_q(\cdot)$).

Asymptotic Hamming bound

Lemma 4.5. *Let $(\mathscr{C}_r)_{r\in\mathbb{N}}$ be a sequence of codes of parameters $[n_r,k_r,d_r]_q$ over a fixed base field $\mathbb{F}_q$ such that n_r tends to infinity. Assume moreover, that the following limits exist*

$$R \stackrel{def}{=} \lim_{r\to+\infty} \frac{k_r}{n_r} \qquad \textit{and} \qquad \delta \stackrel{def}{=} \lim_{r\to+\infty} \frac{d_r}{n_r}.$$

Then,

$$R \leqslant 1 - H_q\left(\frac{\delta}{2}\right).$$

Proof. Let $\varepsilon > 0$ and set for all $r \in \mathbb{N}$, $\delta_r \stackrel{\text{def}}{=} \frac{d_r}{n_r}$ and $R_r \stackrel{\text{def}}{=} \frac{k_r}{n_r}$. From Lemma 3.11(2), for a large enough r, we have

$$\mathrm{Vol}_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right) \geqslant q^{n_r(H_q(\frac{\delta_r}{2}-\frac{1}{2n_r})-\varepsilon)}.$$

Thus, the Hamming bound entails

$$q^{n_r(R_r+H_q(\frac{\delta_r}{2}-\frac{1}{2n_r})-\varepsilon)} \leqslant q^{n_r}.$$

Applying the function $\log_q$ to both sides and dividing them by n_r yields

$$R_r \leqslant 1 - H_q\left(\frac{\delta_r}{2} - \frac{1}{2n_r}\right) + \varepsilon.$$

When r tends to infinity, thanks to the continuity of the entropy function, we get

$$R \leqslant 1 - H_q\left(\frac{\delta}{2}\right) + \varepsilon,$$

which holds for all $\varepsilon > 0$ and yields the result. □

4.1.3 Plotkin bound

Theorem 4.6 (Plotkin bound)**.** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be an $[n,k,d]_q$ code, then*

$$d \leqslant nq^{k-1}\frac{q-1}{q^k-1}.$$

Proof. It is based on a double counting argument. Let $\mathbf{M}$ be a matrix whose rows are *all* the nonzero codewords of $\mathscr{C}$. Thus, $\mathbf{M}$ is a $(q^k-1)\times n$ matrix with entries in $\mathbb{F}_q$. Let A be the number of nonzero entries of $\mathbf{M}$. Clearly, counting the nonzero entries by rows, we get easily that

$$A \geqslant d(q^k-1). \tag{4.1}$$

On the other hand, the i–th column of $\mathbf{M}$ yields the evaluations of a linear form at every element of $\mathscr{C}\setminus\{0\}$. That linear form is either zero on $\mathscr{C}$ or vanishes on a vector subspace of dimension $k-1$. Thus, there are at most q^k-q^{k-1} nonzero entries in every column of $\mathbf{M}$. Consequently,

$$A \leqslant n(q^k-q^{k-1}). \tag{4.2}$$

Combining (4.1) and (4.2) yields the result. □

Theorem 4.7 (Asymptotic Plotkin bound)**.** *Let $(\mathscr{C})_s$ be a sequence of $[n_s,k_s,d_s]_q$ codes over a fixed base field $\mathbb{F}_q$ such that $n_s\to+\infty$ and the sequences $R_s \stackrel{def}{=} \frac{k_s}{n_s}$ and $\delta_s \stackrel{def}{=} \frac{d_s}{n_s}$ converge respectively to reals R,δ. Then,*

$$R \leqslant \max\left\{1-\frac{q\delta}{q-1}\ ,\ 0\right\}.$$

The proof of the Theorem requires first a lemma which involves the code shortening operation (see Definition 1.21).

Lemma 4.8. *Let $(\mathscr{C}_s)_s$ be a sequence of codes over a fixed base field $\mathbb{F}_q$ whose lengths tend to infinity whose rates converge to R and relative distances converge to δ. Then, for all $0<\gamma<1$ there exists a sequence of codes $(\mathscr{C}'_s)_s$ whose rates and relative distances converge respectively to R' and δ' such that,*

$$R' \geqslant R-\gamma \quad\text{and}\quad \delta' \geqslant \min\left\{1,\frac{\delta}{1-\gamma}\right\}.$$

Proof. For all s, choose $I_s\subseteq\{1,\dots,n_s\}$ such that $|I_s|=\lfloor\gamma n_s\rfloor$ and set $\mathscr{C}'_s=\mathcal{S}_{I_s}(\mathscr{C}_s)$. From Proposition 1.24, the parameters of $\mathscr{C}'_s$ satisfy

$$R'_s \geqslant R_s-\gamma \quad\text{and}\quad \delta'_s \geqslant \frac{d'_s}{n_s-\gamma n_s}=\frac{\delta_s}{1-\gamma}. \tag{4.3}$$

Since the sequences $(R'_s)_s$ and $(\delta'_s)_s$ are bounded, then from Bolzano Weierstrass theorem, after replacing $(\mathscr{C}'_s)_s$ by a subsequence, one can assume that the sequences of rates and relative distances converge. Then, passing (4.3) to the limit yields the result. □

Proof of Theorem 4.7. Notice first that, if $d > n\left(\frac{q-1}{q}\right)$, then, the Plotkin bound can be reformulated as follows

$$d(q^k-1)\leqslant nq^k\frac{q-1}{q} \iff q^k\left(d-n\frac{q-1}{q}\right)\leqslant d.$$

In particular,

$$\text{If } d > n\frac{q-1}{q} \text{ then,} \qquad q^k \leqslant \frac{d}{d - n\left(\frac{q-1}{q}\right)}. \tag{4.4}$$

Now consider a sequence of $(\mathscr{C}_s)_s$ of codes of parameters $[n_s, k_s, d_s]_q$ over a fixed field $\mathbb{F}_q$ such that $(n_s)_s$ tends to infinity, $R_s = \frac{k_s}{n_s} \to R$ and $\delta_s = \frac{d_s}{n_s} \to \delta$.

Case 1. If $\delta > \frac{q-1}{q}$, then $\delta_s > \frac{q-1}{q}$ for s large enough and (4.4) gives

$$q^{n_s R_s} \leqslant \frac{\delta_s}{\delta_s - \frac{q-1}{q}}.$$

The right hand side converges to $\frac{\delta}{\delta - \frac{q-1}{q}}$. Therefore, the left hand side is bounded and hence $R = 0$.

Case 2. If $\delta \leqslant \frac{q-1}{q}$. Let $\varepsilon > 0$, and set $\gamma = 1 - \frac{q\delta}{q-1} + \varepsilon$. From Lemma 4.8, there exists a sequence $(\mathscr{C}'_s)_s$ of codes over a fixed field $\mathbb{F}_q$ with asymptotic parameters

$$R' \geqslant R - \gamma \quad \text{and} \quad \delta' \geqslant \frac{\delta}{1-\gamma} = \frac{\delta}{\frac{q\delta}{q-1} - \varepsilon} > \frac{q-1}{q}.$$

Hence, this sequence of codes satisfies the property of Case 1 and hence $R' = 0$, which entails $R \leqslant \gamma$ and hence

$$R \leqslant 1 - \frac{q\delta}{q-1} + \varepsilon.$$

Since the above inequality holds for all $\varepsilon > 0$, we get the result. □

We conclude this section by representing these asymptotic upper bounds for $q = 2$ in Figure 4.1.

4.2 Lower bounds

4.2.1 Gilbert Varshamov bound

The Gilbert Varshamov bound is an existential bound: it asserts the existence of a code whose parameters exceed some bound. This result holds even for non linear codes and it is actually easier to understand in this case. For this reason, we first prove it for nonlinear codes and then provide a proof for the case of linear codes.

Theorem 4.9 (Gilbert Varshamov – the nonlinear case)**.** *Let n, d be two positive integers with $n > d$, then there exists a (possibly nonlinear) code of length n and minimum distance d with M elements and such that:*

$$M\, Vol_q(d, n) \geqslant q^n.$$

Proof. We give a constructive proof by exhibiting an algorithm constructing such a non linear code.

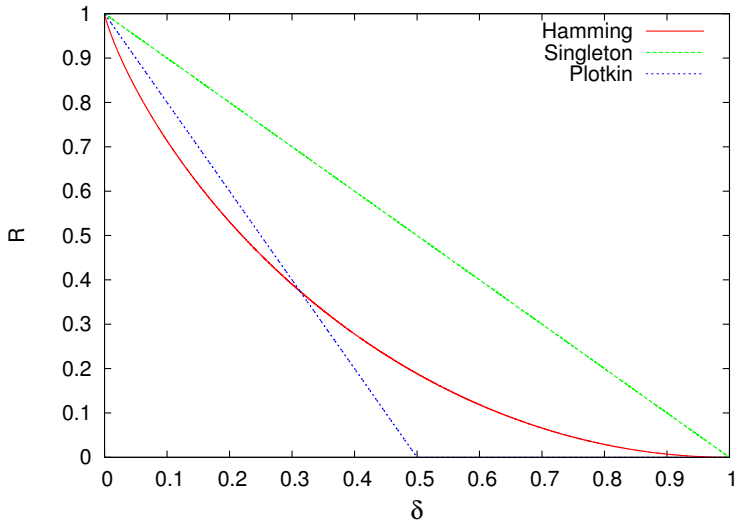


Figure 4.1: Asymptotic upper bounds on codes for $q = 2$

Algorithm 5: Construction of a non linear code of minimum distance bounded below by d

Input : The length n, the prescribed lower bound for the minimum distance d
Output: A code satisfying the Theorem

$\mathscr{C} = \emptyset, \mathcal{U} = \mathbb{F}_q^n$;
while $\mathcal{U} \neq \emptyset$ **do**
  Take $\boldsymbol{c} \in \mathcal{U}$ at random;
  $\mathscr{C} = \mathscr{C} \cup \{\boldsymbol{c}\}, \mathcal{U} = \mathcal{U} \setminus \mathbf{B}_{\mathrm{H}}(\boldsymbol{c}, d)$;
end
Return $\mathscr{C}$;

A loop invariant of the above algorithm is

$$|\mathcal{U}| \geqslant q^n - |\mathscr{C}|\mathrm{Vol}_q(d, n)$$

and the algorithm stops and returns $\mathscr{C}$ when $|\mathcal{U}| = 0$ which entails

$$q^n - |\mathscr{C}|\mathrm{Vol}_q(d, n) \leqslant 0.$$

Moreover, the constructed code has minimum distance d since at each iteration $\mathcal{U}$ is the set of words whose distance to every element of $\mathscr{C}$ is strictly larger than d. □

Remark 25. Actually Theorem 4.9 holds for linear codes, i.e. there exists a linear code satisfying the lower bound of the Theorem. See Exercise 3 Sheet 1.

As for the upper bounds, the Gilbert Varshamov bound has an asymptotic counterpart.

Theorem 4.10. *There exists a sequence of linear codes $(\mathscr{C}_s)_s$ over a fixed field $\mathbb{F}_q$ whose lengths tend to infinity, whose rates sequence converges to R and relative distance sequence converges to δ and such that*

$$R \geqslant 1 - H_q(\delta).$$

Proof. Consider a sequence $(\mathscr{C}_n)_n$ of codes of length n over a fixed field $\mathbb{F}_q$ satisfying the inequality of Theorem 4.9. Then the sequences $(R_n)_n$ and $(\delta_n)_n$ of rates and relative distances are bounded and hence by Bolzano Weierstrass, one can extract a subsequence $(\mathscr{C}_s)_s$ such that the sequences of rates and relative distance converge.

Then, for all s,

$$q^{n_s - k_s} \leqslant \mathrm{Vol}_q(n_s, d_s) \leqslant q^{n_s H_q\left(\frac{d_s}{n_s}\right)},$$

where the second inequality is a consequence of Lemma 3.11. Then, applying $\log_q$ and dividing both sides by n_s yields

$$1 - R_s \leqslant H_q(\delta_s)$$

and we conclude by passing to the limit. □

The asymptotic Gilbert Varshamov bound for $q = 2$ is represented in Figure 4.2.

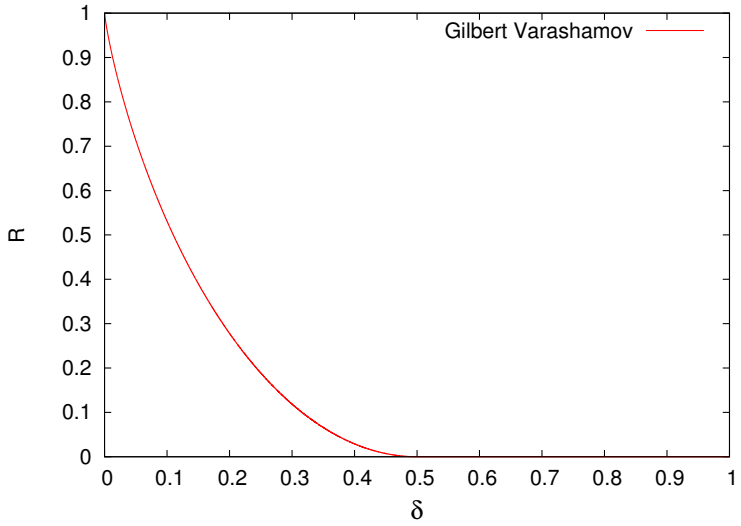


Figure 4.2: The asymptotic Gilbert Varshamov bound for $q = 2$

4.2.2 Random codes and the Gilbert Varshamov bound

A fundamental property of coding theory is that, with a high probability, the parameters of a random code are close to the Gilbert Varshamov bound.

Theorem 4.11. *Let $0 < \delta < 1 - \frac{1}{q}$. Let $\varepsilon > 0$ and let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a random code of dimension $k \leqslant (1 - H_q(\delta) - \varepsilon)n$. Then,*

$$\mathbb{P}\big(d_{\min}(\mathscr{C}) > \delta n\big) \geqslant 1 - q^{-\varepsilon n}.$$

Proof. From Lemma 3.12, $\mathscr{C} = \mathrm{Im}(\boldsymbol{G})$ where $\boldsymbol{G}$ is a uniformly random $k \times n$ matrix of rank k. Then,

$$\mathbb{P}\big(d_{\min}(\mathscr{C}) \leqslant \delta n\big) = \mathbb{P}\big(\exists \boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\},\ \mathrm{w_H}(\boldsymbol{mG}) \leqslant \delta n\big) \tag{4.5}$$

$$\leqslant \sum_{\boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\}} \mathbb{P}\big(\mathrm{w_H}(\boldsymbol{mG}) \leqslant \delta n\big). \tag{4.6}$$

From Lemma 3.13, for all $\boldsymbol{m} \in \mathbb{F}_q^k \setminus \{0\}$ the word $\boldsymbol{mG}$ is a uniformly random word of $\mathbb{F}_q^n \setminus \{0\}$. Therefore,

$$\mathbb{P}\big(\mathrm{w_H}(\boldsymbol{mG}) \leqslant \delta\big) = \frac{\mathrm{Vol}_q(\delta n, n)}{q^n} \leqslant q^{n(H_q(\delta)-1)},$$

where the last inequality is a direct consequence of Lemma 3.11. Therefore, (4.6) becomes

$$\begin{aligned}\mathbb{P}\big(d_{\min}(\mathscr{C}) \leqslant \delta n\big) &\leqslant (q^k - 1) q^{n(H_q(\delta)-1)} \\ &\leqslant q^{n(1-H_q(\delta)-\varepsilon)-n(H_q(\delta)-1)} \\ &\leqslant q^{-\varepsilon n}.\end{aligned}$$

□

4.3 Conclusion

Finally a natural question is *which pair (δ, R) is achievable by a sequence of codes?* That is, for which pairs (δ, R) there exist a sequence of codes $(\mathscr{C}_s)_s$ whose relative distance sequence converges to δ and rate sequence converges to R? is the limit of the re To address this question, we use first the following lemmas allowing to construct bad codes from good ones.

Lemma 4.12. *Let $(\mathscr{C}_s)_s$ be a sequence of codes over a fixed field $\mathbb{F}_q$ whose rates and relative distances converge to (δ, R), then, for all $0 \leqslant R' \leqslant R$, there exists a sequence $(\mathscr{C}'_s)_s$ whose relative parameters converge to (R', δ).*

Proof. For all $s \in \mathbb{N}$ choose a minimum weight codeword $\boldsymbol{c}_s \in \mathscr{C}_s$. Then, choose an arbitrary subcode $\mathscr{C}'_s$ of $\mathscr{C}_s$ containing $\boldsymbol{c}_s$ and of dimension $\min\{\lfloor R' n_s \rfloor, \dim \mathscr{C}_s\}$. For s large enough, $\dim \mathscr{C}'_s = \lfloor R' n_s \rfloor$ and its minimum distance equals that of $\mathscr{C}_s$ since it contains $\boldsymbol{c}_s$. Therefore, the parameters of the sequence $(\mathscr{C}'_s)_s$ converge to (R', δ). □

Lemma 4.13. *Let $(\mathscr{C}_s)_s$ be a sequence of codes over a fixed field $\mathbb{F}_q$ whose rates and relative distances converge to (δ, R), then, for all $0 \leqslant \delta' \leqslant \delta$, there exists a sequence $(\mathscr{C}'_s)_s$ whose relative parameters converge to (R, δ').*

Proof. For all s, choose a minimum weight codeword $\boldsymbol{c}_s$ of $\mathscr{C}_s$. For s large enough, $\mathrm{w_H}(\boldsymbol{c}_s) > \delta' n_s$ and, for such large enough n_s choose $I_s \subseteq \{1, \ldots, n_s\}$ such that

$$|I_s| = \left\lfloor \frac{\mathrm{w_H}(\boldsymbol{c}_s) - \delta' n_s}{1 - \delta'} \right\rfloor$$

and which is contained in the *support* of $\boldsymbol{c}_s$, i.e. for all $i \in I_s$, the i–th entry of $\boldsymbol{c}_s$ is nonzero. Such a choice is possible since one checks easily that $\mathrm{w_H}(\boldsymbol{c}_s) \geqslant |I_s|$. Notice that

$$\lim_{s\to+\infty} \frac{|I_s|}{n_s} = \frac{\delta - \delta'}{1-\delta'}. \tag{4.7}$$

Next, consider the map

$$\phi_s : \left\{ \begin{array}{ccc} \mathbb{F}_q^{n_s} & \longrightarrow & \mathbb{F}_q^{n_s - |I_s|} \\ (x_1, \ldots, x_{n_s}) & \longmapsto & (x_i)_{i \in \{1,\ldots,n_s\} \setminus I_s} \end{array} \right.$$

and set $\mathscr{C}_s'' \stackrel{\text{def}}{=} \phi_s(\mathscr{C}_s)$ and $\boldsymbol{c}_s'' \stackrel{\text{def}}{=} \phi_s(\boldsymbol{c}_s)$. This code has length $n_s - |I_s|$ and, by construction, $\boldsymbol{c}_s''$ is a minimum weight codeword of $\mathscr{C}_s$ of weight $\mathrm{w_H}(\boldsymbol{c}_s) - |I_s|$. Moreover, since $|I_s|$ is smaller than the minimum distance of $\mathscr{C}_s$, then the restriction of ϕ_s to $\mathscr{C}_s$ is injective (same argument as in the proof of Singleton bound, Theorem 4.1), hence $\mathscr{C}_s$ and $\mathscr{C}_s''$ have the same dimension. Therefore since $\mathscr{C}_s''$ has a shorter length, its rate is higher than that of $\mathscr{C}_s$.

Finally, choose $\mathscr{C}_s'$ a subcode of $\mathscr{C}_s''$ of dimension $\min\{\lfloor R(n_s - |I_s|)\rfloor, \dim \mathscr{C}_s''\}$ and containing $\boldsymbol{c}_s''$ which equals $\lfloor R(n_s - |I_s|)\rfloor$ for s large enough. Hence, for s large enough, the code $\mathscr{C}_s'$ has length $n_s - |I_s|$, dimension $\lfloor R(n_s - |I_s|)\rfloor$ and minimum distance $\mathrm{w_H}(\boldsymbol{c}_s'') = \mathrm{w_H}(\boldsymbol{c}_s) - |I_s|$. Therefore, the rate of the sequence $(\mathscr{C}_s)_s$ converges to R and, thanks to (4.7), the relative distance converges to

$$\lim_{s\to+\infty} \frac{\mathrm{w_H}(\boldsymbol{c}_s) - |I_s|}{n_s - |I_s|} = \lim_{s\to+\infty} \frac{\frac{\mathrm{w_H}(\boldsymbol{c}_s)}{n_s} - \frac{|I_s|}{n_s}}{1 - \frac{|I_s|}{n_s}} = \frac{\delta - \frac{\delta-\delta'}{1-\delta'}}{1 - \frac{\delta-\delta'}{1-\delta'}} = \delta'.$$

□

Consequently, Lemmas 4.12 and 4.13 assert that, as soon as a point (δ, R) is achieved, every point in the square with top right hand corner (δ, R) and bottom left hand corner $(0, 0)$ can be achieved too. Thanks to this fact and to Theorem 4.11, we know that all the area below the Gilbert Varshamov bound is achievable. On the other hand, we know that every point above the Hamming, or Plotkin bound is non achievable. This raises the question of the intermediary area represented in Figure 4.3. *Are some points above the Gilbert Varshamov bound achievable?*

This problem is still widely open. In particular, for binary codes, there is no known explicit family beating the Gilbert Varshamov bound and there is no evidence about the existence or non existence of such a family. Up to the beginning of the 80's the common belief was that asymptotic Gilbert Varshamov bound is optimal and no family could exceed it. This belief turns out to be false. Indeed, in 1982, Tsfasman Vlăduţ and Zink proved the existence of families of codes beating Gilbert Varshamov bound over all field $\mathbb{F}_q$ such that q is a square and $q \geqslant 49$. These codes constructed by methods arising from algebraic geometry and number theory. For further details see [VNT07, TVZ82, Mor91].

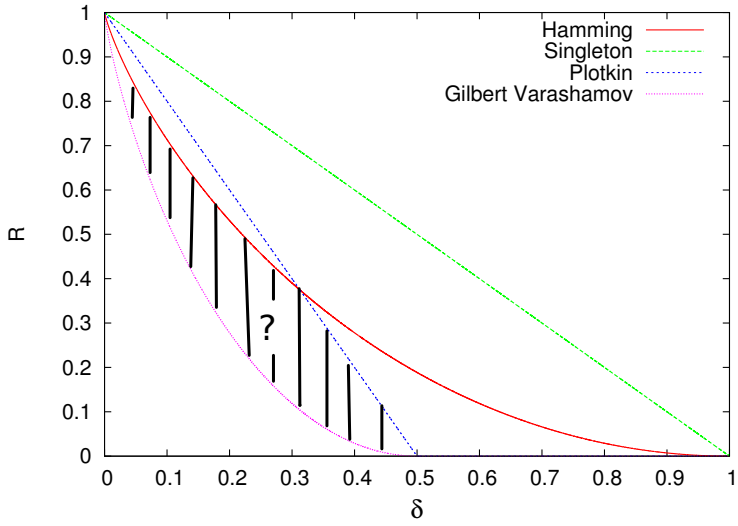


Figure 4.3: An open Problem (the bounds are represented for $q = 2$)

Chapter 5

Duality

Duality is a fundamental notion in coding theory. Roughly speaking, to each code $\mathscr{C} \subseteq \mathbb{F}_q^n$ is associated a *dual* or *orthogonal* code $\mathscr{C}^\perp$. In many situations of coding theory, the properties of the dual code help to have a better understanding of the code itself.

5.1 The dual code

5.1.1 Definitions

We introduce the *canonical inner product* on $\mathbb{F}_q^n$ defined as

$$\langle \cdot, \cdot \rangle : \left\{ \begin{array}{ccc} \mathbb{F}_q^n \times \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q \\ (\boldsymbol{x}, \boldsymbol{y}) & \longmapsto & \sum_{i=1}^n x_i y_i \end{array} \right. .$$

Proposition 5.1. *The canonical inner product is non degrenerated, i.e. if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$ for all $\boldsymbol{y} \in \mathbb{F}_q^n$, then $\boldsymbol{x} = 0$.*

Proof. Let $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)$ be the canonical basis of $\mathbb{F}_q^n$. Then, for all $i \in \{1, \ldots, n\}$,

$$\langle \boldsymbol{x}, \boldsymbol{e}_i \rangle = x_i$$

Thus, if $\langle \boldsymbol{x}, \boldsymbol{e}_i \rangle = 0$ for all i, then $x_i = 0$ for all i. □

Definition 5.2. Let $\mathscr{C} \subseteq \mathbb{F}_q^n$. The *orthogonal* or *dual* of $\mathscr{C}$ is defined as

$$\mathscr{C}^\perp \stackrel{\text{def}}{=} \left\{ \boldsymbol{x} \in \mathbb{F}_q^n \mid \forall \boldsymbol{c} \in \mathscr{C}, \langle \boldsymbol{c}, \boldsymbol{x} \rangle = 0 \right\}.$$

A natural question is *why choosing this bilinear form?* There are many non degenerated bilinear forms on $\mathbb{F}_q^n$, why choosing this one? The reason is that this bilinear form behaves well with the Hamming metric:

- The canonical basis is orthogonal for $\langle \cdot, \cdot \rangle$ and the Hamming weight is naturally associated to the canonical basis since the Hamming weight of a word is the number of terms of its decomposition in this basis.

- One can prove (see Exercises) that the linear isometries of $\mathbb{F}_q^n$ for the Hamming distance form a group of matrices spanned by the permutation matrices and the invertible diagonal matrices. For this bilinear form, the diagonal matrices are self-adjoint (i.e. symmetric) and the permutation matrices are orthogonal, i.e. for such a permutation σ,
$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n,\ \langle \boldsymbol{x}, \sigma(\boldsymbol{y}) \rangle = \left\langle \sigma^{-1}(\boldsymbol{x}), \boldsymbol{y} \right\rangle .$$
Therefore, Hamming isometries are *remarkable* endomorphisms with respect of $\langle \cdot, \cdot \rangle$.

5.1.2 Basic properties

Notation 5.1. In what follows, given a matrix $\boldsymbol{M} \in \mathfrak{M}_{a,b}(\mathbb{F}_q)$, we denote respectively by $\ker_l \boldsymbol{M}$ and $\ker_r \boldsymbol{M}$ its left and right kernel. That is

$$\ker_l \boldsymbol{M} \stackrel{\text{def}}{=} \left\{ \boldsymbol{m} \in \mathbb{F}_q^a \mid \boldsymbol{m}\boldsymbol{M} = 0 \right\} \quad \text{and} \quad \ker_r \boldsymbol{M} \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{F}_q^b \mid \boldsymbol{M}\boldsymbol{v}^T = 0 \right\} .$$

In the same manner, we define left and right images as

$$\mathrm{Im}_l \boldsymbol{M} \stackrel{\text{def}}{=} \left\{ \boldsymbol{m}\boldsymbol{M} \mid \boldsymbol{m} \in \mathbb{F}_q^a \right\} \quad \text{and} \quad \mathrm{Im}_r \boldsymbol{M} \stackrel{\text{def}}{=} \left\{ \boldsymbol{M}\boldsymbol{v}^T \mid \boldsymbol{v} \in \mathbb{F}_q^b \right\} .$$

Proposition 5.3. *Let $\boldsymbol{G}$, $\boldsymbol{H}$ be respectively a generator matrix and a parity–check matrix of $\mathscr{C}$, then*

- *$\boldsymbol{H}$ is a generator matrix for $\mathscr{C}^\perp$;*
- *$\boldsymbol{G}$ is a parity–check matrix for $\mathscr{C}^\perp$.*

In particular,
$$\dim \mathscr{C} + \dim \mathscr{C}^\perp = n.$$

Proof. By definition, $\mathscr{C} = \ker_r \boldsymbol{H}$. Therefore, every row $\boldsymbol{h}$ of $\boldsymbol{H}$ satisfies $\langle \boldsymbol{h}, \boldsymbol{c} \rangle = 0$ for all $\boldsymbol{c} \in \mathscr{C}$. Thus, $\mathrm{Im}_r \boldsymbol{H} \subseteq \mathscr{C}^\perp$. This entails in particular $\dim \mathscr{C}^\perp \geqslant n - k$. On the other hand, $\mathscr{C} = \mathrm{Im}_l \boldsymbol{G}$ and hence every row $\boldsymbol{g}$ of $\boldsymbol{G}$ is in $\mathscr{C}$ and satisfies $\langle \boldsymbol{g}, \boldsymbol{c} \rangle = 0$ for all $\boldsymbol{c} \in \mathscr{C}^\perp$. Therefore, $\mathscr{C}^\perp \subseteq \ker_r \boldsymbol{G}$ which entails in particular that $\dim \mathscr{C}^\perp \leqslant n - k$. Putting all together, we get that $\dim \mathscr{C}^\perp = n - k$ and $\mathscr{C}^\perp = \ker_r \boldsymbol{G} = \mathrm{Im}_l \boldsymbol{H}$. □

Proposition 5.4. *(i)* $(\mathscr{C}^\perp)^\perp = \mathscr{C}$.

(ii) $\{0\}^\perp = \mathbb{F}_q^n$ *and* $(\mathbb{F}_q^n)^\perp = \{0\}$.

(iii) Given two codes $\mathscr{C}, \mathscr{C}'$ such that $\mathscr{C} \subseteq \mathscr{C}'$, then $\mathscr{C}^\perp \supseteq \mathscr{C}'^\perp$.

(iv) Given two codes $\mathscr{C}, \mathscr{C}'$, $(\mathscr{C} + \mathscr{C}')^\perp = \mathscr{C}^\perp \cap \mathscr{C}'^\perp$ and $(\mathscr{C} \cap \mathscr{C}')^\perp = \mathscr{C}^\perp + \mathscr{C}'^\perp$.

Proof. Let $\boldsymbol{c} \in \mathscr{C}$, then, for all $\boldsymbol{c}' \in \mathscr{C}^\perp$, $\langle \boldsymbol{c}, \boldsymbol{c}' \rangle$ and hence $\boldsymbol{c} \in (\mathscr{C}^\perp)^\perp$. Moreover, from Proposition 5.3, $\mathscr{C}$ and $(\mathscr{C}^\perp)^\perp$ have the same dimension. This proves (i). The equality $\{0\}^\perp = \mathbb{F}_q^n$ is obvious. The equality $(\mathbb{F}_q^n)^\perp = \{0\}$ is a consequence of (i). This proves (ii). Let $\boldsymbol{c}' \in \mathscr{C}'^\perp$, then by definition, for all $\boldsymbol{c} \in \mathscr{C}'$, $\langle \boldsymbol{c}', \boldsymbol{c} \rangle = 0$. Since $\mathscr{C} \subseteq \mathscr{C}'$, for all $\boldsymbol{c} \in \mathscr{C}$, $\langle \boldsymbol{c}', \boldsymbol{c} \rangle$, hence $\boldsymbol{c}' \in \mathscr{C}^\perp$, this proves (iii). Finally, let since $\mathscr{C}, \mathscr{C}' \subseteq \mathscr{C} + \mathscr{C}'$, then, from (iii), $(\mathscr{C} + \mathscr{C}')^\perp \subseteq \mathscr{C}^\perp \cap \mathscr{C}'^\perp$. Conversely, let $\boldsymbol{c} \in \mathscr{C}^\perp \cap \mathscr{C}'^\perp$ then, for all $\boldsymbol{x} \in \mathscr{C}$ and all $\boldsymbol{x}' \in \mathscr{C}'$, we have $\langle \boldsymbol{c}, \boldsymbol{x} + \boldsymbol{x}' \rangle = \langle \boldsymbol{c}, \boldsymbol{x} \rangle + \langle \boldsymbol{c}, \boldsymbol{x}' \rangle = 0$ and hence $\mathscr{C}^\perp \cap \mathscr{C}'^\perp \subseteq (\mathscr{C} + \mathscr{C}')^\perp$. This proves the first equality of (iv). The seconde equality is obtained by replacing $\mathscr{C}$ by $\mathscr{C}^\perp$, $\mathscr{C}'$ by $\mathscr{C}'^\perp$ and applying (i). □

5.1.3 Caution

Let us emphasize two facts which may be confusing.

Be careful with the term "dual"

Regarding a code $\mathscr{C}$ as an intrinsic vector space of dimension k, what is usually referred to as the dual of $\mathscr{C}$ in linear algebra is the space of linear forms on $\mathscr{C}$, that is the space:

$$\mathscr{C}^\vee \stackrel{\text{def}}{=} \mathrm{Hom}_{\mathbb{F}_q}(\mathscr{C}, \mathbb{F}_q).$$

This vector space is not the dual code $\mathscr{C}^\perp$. Indeed, this space has dimension k while $\mathscr{C}^\perp$ has dimension $n - k$ as proved in Proposition 5.3. If we used the classical terminology of Euclidean algebra, the space $\mathscr{C}^\perp$ would be referred to as the *orthogonal* space of $\mathscr{C}$.

Be careful with isotropy

It is tantalising to reason by analogy with Euclidean geometry, since $\langle \cdot, \cdot \rangle$ looks like the canonical scalar product over the reals. Moreover, Proposition 5.4 suggests that many properties from Euclidean geometry still hold in our context.

However, we should be careful with the following fact. Given an Euclidean or Hermitian vector space E and $F \subseteq E$, then we always have $F \oplus F^\perp = E$, which is false in positive characteristic: in general given a code $\mathscr{C} \subseteq \mathbb{F}_q^n$ we may have

$$\mathscr{C} \cap \mathscr{C}^\perp \neq \{0\}.$$

The intersection $\mathscr{C} \cap \mathscr{C}^\perp$ is usually referred to as the *hull* of the code and may be large. An extremal example is given by the so called *self–dual codes* which satisfy $\mathscr{C} = \mathscr{C}^\perp$. For instance the code over $\mathbb{F}_2$ with generator matrix

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

This difference with Euclidean geometry is due to the presence of nonzero isotropic vectors for the quadratic form associated to the inner product i.e. nonzero vectors satisfying $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$. Such vectors do not exist in an Euclidean or Hermitian vector space while one can prove that $\mathbb{F}_q^n$ has nonzero isotropic vectors for all $n \geqslant 3$.

5.2 Duality between some code constructions

5.2.1 Duality between shortening and pucturing

Theorem 5.5. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code and $I \subseteq \{1, \ldots, n\}$. Then*

$$\mathcal{S}_I\left(\mathscr{C}^\perp\right) = \left(\mathcal{P}_I\left(\mathscr{C}\right)\right)^\perp \qquad \text{and, equivalently,} \qquad \mathcal{P}_I\left(\mathscr{C}^\perp\right) = \left(\mathcal{S}_I\left(\mathscr{C}\right)\right)^\perp.$$

Proof. Let $\boldsymbol{G}$ be a full rank generator matrix of $\mathscr{C}$ and $\boldsymbol{G}'$ be the matrix obtained from $\boldsymbol{G}$ by deleting the columns whose index is in I. From Proposition 5.3, $\boldsymbol{G}$ is a parity–check matrix for $\mathscr{C}^\perp$, from Proposition 1.19, $\boldsymbol{G}'$ is a generator matrix of $\mathcal{P}_I\left(\mathscr{C}\right)$ and, from Proposition 1.23, it is a parity–check matrix of $\mathcal{S}_I\left(\mathscr{C}^\perp\right)$. Appying Proposition 5.3 we get that $\left(\mathcal{P}_I\left(\mathscr{C}\right)\right)^\perp = \mathcal{S}_I\left(\mathscr{C}^\perp\right)$.

The second statement can be obtained directly by replacing $\mathscr{C}$ by $\mathscr{C}^\perp$ and using $\left(\mathscr{C}^\perp\right)^\perp = \mathscr{C}$ (Proposition 5.4(i)) □

5.2.2 Duality between subfield subcode and trace code : Delsarte Theorem

In what follows we deal with codes over $\mathbb{F}_q$ and codes over $\mathbb{F}_{q^m}$. Thus, we consider two canonical inner products, one on $\mathbb{F}_q^n$ and another one on $\mathbb{F}_{q^m}^n$ which are respectively denoted as $\langle \cdot, \cdot \rangle_{\mathbb{F}_q^n}$ and $\langle \cdot, \cdot \rangle_{\mathbb{F}_{q^m}^n}$. Note that there is a canonical field inclusion $\mathbb{F}_q \hookrightarrow \mathbb{F}_{q^m}$ and hence that given two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}_q^n$ then

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_q^n} = \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}^n}.$$

Theorem 5.6 (Delsarte Theorem [Del75, Theorem 2])**.** *Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$. Then,*

$$\left(\mathscr{C}_{|\mathbb{F}_q}\right)^\perp = \mathit{Tr}(\mathscr{C}^\perp) \quad \textit{and, equivalently} \quad \left(\mathscr{C}^\perp\right)_{|\mathbb{F}_q} = \left(\mathit{Tr}(\mathscr{C})\right)^\perp$$

Proof. As for the proof of Theorem 5.5, the two statements are equivalent and one deduces one from the other by replacing $\mathscr{C}$ by $\mathscr{C}^\perp$ and using Proposition 5.4(i). Thus, let us prove the first statement. Let $\boldsymbol{u} \in \mathscr{C}^\perp$ and $\boldsymbol{v} \in \mathscr{C}_{|\mathbb{F}_q}$.

$$\langle \mathrm{Tr}(\boldsymbol{u}), \boldsymbol{v} \rangle_{\mathbb{F}_q^n} = \sum_{i=1}^{n} \mathrm{Tr}(u_i) v_i$$

and, since for all i, $v_i \in \mathbb{F}_q$, using the $\mathbb{F}_q$–linearity of the trace we obtain.

$$\begin{aligned} \langle \mathrm{Tr}(\boldsymbol{u}), \boldsymbol{v} \rangle_{\mathbb{F}_q^n} &= \sum_{i=1}^{n} \mathrm{Tr}(u_i v_i) \\ &= \mathrm{Tr}(\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}^n}) = 0. \end{aligned}$$

This proves

$$\mathrm{Tr}(\mathscr{C}^\perp) \subseteq (\mathscr{C}_{|\mathbb{F}_q})^\perp. \tag{5.1}$$

Conversely, let $\boldsymbol{u} \in (\mathrm{Tr}(\mathscr{C}^\perp))^\perp$ and $\boldsymbol{v} \in \mathscr{C}^\perp$. Let $\lambda \in \mathbb{F}_{q^m}$.

$$\begin{aligned}\mathrm{Tr}(\lambda \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}^n}) &= \mathrm{Tr}\left(\lambda \sum_{i=1}^n u_i v_i\right) \\ &= \sum_{i=1}^n \mathrm{Tr}(\lambda u_i v_i).\end{aligned}$$

Moreover, since $\boldsymbol{u} \in (\mathrm{Tr}(\mathscr{C}^\perp))^\perp$, it is an element of $\mathbb{F}_q^n$ and hence, by the $\mathbb{F}_q$–linearity of the trace, we get

$$\begin{aligned}\mathrm{Tr}(\lambda \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}^n}) &= \sum_{i=1}^n u_i \mathrm{Tr}(\lambda v_i) \\ &= \langle \boldsymbol{u}, \lambda \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}^n}.\end{aligned}$$

Finally, since $\mathscr{C}^\perp$ is $\mathbb{F}_{q^m}$–linear, $\lambda v \in \mathscr{C}^\perp$ and the last term is zero. Thus,

$$\forall \lambda \in \mathbb{F}_{q^m}, \quad \mathrm{Tr}(\lambda \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}}) = 0.$$

Consequently, from Corollary 1.29, we get $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{F}_{q^m}^n} = 0$. Therefore,

$$(\mathrm{Tr}(\mathscr{C}^\perp))^\perp \subseteq \mathscr{C}$$

and, since the left–hand term is a code contained in $\mathbb{F}_q^n$, we get

$$(\mathrm{Tr}(\mathscr{C}^\perp))^\perp \subseteq \mathscr{C} \cap \mathbb{F}_q^n = \mathscr{C}_{|\mathbb{F}_q}$$

and hence, using Proposition 5.4(iii) we conclude that

$$(\mathscr{C}_{|\mathbb{F}_q})^\perp \subseteq \mathrm{Tr}(\mathscr{C}^\perp). \tag{5.2}$$

Putting (5.2) and (5.2) together, we get the result. □

5.3 Metric relations between a code and its dual, McWilliams Theorem

In the previous section, we obtained an elementary identity relating the dimension of a code with that of its dual, namely

$$\dim \mathscr{C} + \dim \mathscr{C}^\perp = n.$$

A natural question, is *is there a relation between the minimum distance of $\mathscr{C}$ and that of $\mathscr{C}^\perp$?* Unfortunately, the answer is negative. It is possible to construct sequences of codes $(\mathscr{C}_s)_s$ whose minimum distances and dual minimum distances are linear in the code length, i.e. satisfy

$$d_s \sim \delta n_s \quad \text{and} \quad d_s^\perp \sim \delta^\perp n_s$$

with $\delta, \delta^\perp > 0$. This is for instance what happens for the so-called *algebraic geometry codes* constructed by Tsfasman Vlăduţ and Zink to beat the asymptotic Gilbert Varshamov bound (see Chapter 4). On the other hand, some families of codes called *LDPC codes*[1] have a minimum distance linear in the length and a bounded dual distance, i.e:

$$d_s \sim \delta n_s \quad \text{and} \quad d_s^\perp \leqslant w$$

for some fixed constant w and for some $\delta > 0$.

Actually, the previous examples show that the minimum distance is a too coarse invariant for this question. On the other hand, there exist actual deep relations between the metric structure of a code and its dual. But these relations imply not only the minimum distance but all the weights of the codewords.

5.3.1 The weight enumerator of a code

Definition 5.7. Let $\mathscr{C}$ be a code, then for all $t \in \{0, \ldots, n\}$, we define

$$w_t(\mathscr{C}) \stackrel{\text{def}}{=} |\{\boldsymbol{c} \in \mathscr{C} \mid \mathrm{w_H}(\boldsymbol{c}) = t\}|.$$

The *weight enumerator* (resp. *homogeneous weight enumerator*) is the polynomial defined as

$$P_{\mathscr{C}}^\sharp(z) \stackrel{\text{def}}{=} \sum_{i=0}^{n} w_i z^i \quad \text{and} \quad P_{\mathscr{C}}(x, y) = \sum_{i=0}^{n} w_i x^i y^{n-i}.$$

Remark 26. Since $\mathscr{C}$ is linear, it contains 0 and hence $P_{\mathscr{C}}^\sharp(0) = 1$. Moreover, if $\mathscr{C}$ has minimum distance d, then

$$P_{\mathscr{C}}^\sharp(z) = 1 + z^d Q(z)$$

for some polynomial Q with $\deg Q \leqslant n - d$.

5.3.2 McWilliams Theorem

Theorem 5.8 (McWilliams Theorem)**.** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a linear code, then*

$$P_{\mathscr{C}^\perp}(x, y) = \frac{1}{|\mathscr{C}|} P_{\mathscr{C}}(y - x, y + (q-1)x).$$

[1] Low Density Parity Check codes, i.e. codes having a "sparse" parity check matrice. More precisely, a sequence of codes $(\mathscr{C}_s)_s$ is said to be LDPC if they have for all s a parity–check matrix $\boldsymbol{H}_s$ with row weight bounded by a constant w.

This statement is proved in §5.3.3.

Example 5.9. Let $\mathscr{C} \subseteq \mathbb{F}_2^n$ be the repetition code i.e. the code of dimension 1 spanned by $(1, \dots, 1)$. We have

$$P_{\mathscr{C}}(x, y) = y^n + x^n.$$

The dual $\mathscr{C}^\perp$ is the parity code (see §1.3.2). Thus,

$$P_{\mathscr{C}^\perp} = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} x^{2i} y^{n-2i}.$$

Moreover,

$$\begin{aligned} P_{\mathscr{C}}(y - x, y + x) &= (y - x)^n + (y + x)^n \\ &= \sum_{k=0}^{n} \binom{n}{k} (-1)^k x^k y^{n-k} + \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} \\ &= 2 \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} x^{2i} y^{n-2i}. \end{aligned}$$

Which gives $\frac{1}{2} P_{\mathscr{C}}(y - x, y + x) = P_{\mathscr{C}^\perp}(x, y)$.

Corollary 5.10 (McWilliams Theorem for the non homogeneous weight enumerator)**.** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a linear code, then*

$$P^\sharp_{\mathscr{C}^\perp}(z) = \frac{1}{|\mathscr{C}|} (1 + (q-1)z)^n P^\sharp_{\mathscr{C}} \left(\frac{1 - z}{1 + (q-1)z} \right).$$

Proof of Corollary 5.10. It is easy to deduce $P^\sharp_{\mathscr{C}}$ from $P_{\mathscr{C}}$:

$$P^\sharp_{\mathscr{C}}(z) = P_{\mathscr{C}}(z, 1).$$

On the other hand, since $P_{\mathscr{C}}$ is a homogeneous polynomial of degree n, then for all $\lambda \in \mathbb{F}_q$,

$$P_{\mathscr{C}}(\lambda x, \lambda y) = \lambda^n P_{\mathscr{C}}(x, y).$$

Therefore,

$$P_{\mathscr{C}}(x, y) = y^n P_{\mathscr{C}} \left(\frac{x}{y}, 1 \right) = y^n P^\sharp_{\mathscr{C}} \left(\frac{x}{y} \right). \tag{5.3}$$

Thus, set $z = \frac{x}{y}$, then

$$y^n \left(1 + (q-1)\frac{x}{y} \right)^n P^\sharp_{\mathscr{C}} \left(\frac{1 - \frac{x}{y}}{1 + (q-1)\frac{x}{y}} \right) = (y + (q-1)x) P^\sharp_{\mathscr{C}} \left(\frac{y - x}{y + (q-1)x} \right)$$

and, from (5.3), the last expression is nothing but $P_{\mathscr{C}}(y - x, y + (q-1)x)$. □

5.3.3 Proof of McWilliams Theorem

Trace and characters

The proof of McWilliams requires the introduction of the characters of the additive group $\mathbb{F}_q$.

Notation 5.2. In what follows, $\zeta_n \in \mathbb{C}$ denotes the primitive p–th root of 1, $\zeta_p \stackrel{\text{def}}{=} e^{\frac{2i\pi}{p}}$. Since $\zeta_p^p = 1$, then, for all $a \in \mathbb{Z}$ the number ζ_p^a depends only on the class of a modulo p. Therefore, for all $u \in \mathbb{F}_p$, the expression ζ_p^u makes sense.

Definition 5.11 (Characters of $\mathbb{F}_q$)**.** For all $a \in \mathbb{F}_q$, the *character* χ_a is defined as the map

$$\chi_a : \left\{ \begin{array}{ccc} \mathbb{F}_q & \longrightarrow & \mathbb{C}^\times \\ x & \longmapsto & \zeta_p^{\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(ax)} \end{array} \right. .$$

In particular, if $q = p$, then $\chi_a(x) = \zeta_p^{ax}$.

Sums of characters

The following statements are central in the proof of Theorem 5.8.

Lemma 5.12. *(i) Let $x_0 \in \mathbb{F}_q$, then*

$$\sum_{a \in \mathbb{F}_q} \chi_a(x_0) = \left\{ \begin{array}{ll} 0 & \text{if} \quad x_0 \neq 0 \\ q & \text{else.} \end{array} \right.$$

(ii) Let $a_0 \in \mathbb{F}_q$, then

$$\sum_{x \in \mathbb{F}_q} \chi_{a_0}(x) = \left\{ \begin{array}{ll} 0 & \text{if} \quad a_0 \neq 0 \\ q & \text{else.} \end{array} \right.$$

Proof. Let us prove first that both statements are equivalent, indeed,

$$\sum_{a \in \mathbb{F}_q} \chi_a(x_0) = \sum_{a \in \mathbb{F}_q} \zeta_p^{\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(ax_0)} = \sum_{x \in \mathbb{F}_q} \zeta_p^{\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(a_0 x)} = \sum_{x \in \mathbb{F}_q} \chi_{a_0}(x).$$

Therefore, it is sufficient to prove (i). If $x_0 = 0$, then the sum becomes

$$\sum_{a \in \mathbb{F}_q} \zeta_p^0 = |\mathbb{F}_q| = q.$$

If $x_0 \neq 0$, then the map

$$\mathrm{Tr}(\,\cdot\, x_0) : \left\{ \begin{array}{ccc} \mathbb{F}_q & \longrightarrow & \mathbb{F}_p \\ a & \longmapsto & \mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(ax_0) \end{array} \right.$$

is a nonzero $\mathbb{F}_p$–linear form on $\mathbb{F}_q$ (see Lemma 1.28). Hence, its kernel is an $\mathbb{F}_p$–hyperplane of $\mathbb{F}_q$ and hence has $\mathbb{F}_p$–dimension $m-1$, that is

$$|\ker \mathrm{Tr}(\,\cdot\, x_0)| = p^{m-1}.$$

More generally, since it is nonzero, then it is surjective and for all $u \in \mathbb{F}_p$,

$$|\{a \in \mathbb{F}_q \mid \mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(ax_0) = u\}| = |\ker \mathrm{Tr}(\,\cdot\, x_0)| = p^{m-1}.$$

Consequently,

$$\sum_{a\in\mathbb{F}_q} \zeta_p^{\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(ax_0)} = p^{m-1}\sum_{u\in\mathbb{F}_p}\zeta^u = p^{m-1}\sum_{j=0}^{p-1}\zeta_p^j = p^{m-1}\frac{1-\zeta_p^p}{1-\zeta_p} = 0$$

□

Lemma 5.13. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code, then for all $\boldsymbol{u} \in \mathbb{F}_q^n$*

$$\sum_{\boldsymbol{c}\in\mathscr{C}} \chi_1(\langle \boldsymbol{c}, \boldsymbol{u}\rangle) = \begin{cases} |\mathscr{C}| & \text{if} \quad \boldsymbol{u} \in \mathscr{C}^\perp \\ 0 & \text{else.} \end{cases}$$

Proof. If $\boldsymbol{u} \in \mathscr{C}^\perp$ then, the sum is

$$\sum_{\boldsymbol{c}\in\mathscr{C}} \chi_1(0) = |\mathscr{C}|.$$

Now, assume that $\boldsymbol{u} \notin \mathscr{C}^\perp$. The map

$$\langle\,\cdot\,, \boldsymbol{u}\rangle : \begin{cases} \mathscr{C} & \longrightarrow & \mathbb{F}_q \\ \boldsymbol{c} & \longmapsto & \langle \boldsymbol{c}, \boldsymbol{u}\rangle \end{cases}$$

is an $\mathbb{F}_q$–linear form over $\mathscr{C}$. Moreover, by definition of $\mathscr{C}^\perp$, since we assumed $\boldsymbol{u} \notin \mathscr{C}^\perp$, this linear form is nonzero. Therefore, its kernel is a hyperplane of $\mathscr{C}$ and hence has q^{k-1} elements, where k denotes the dimension of $\mathscr{C}$. Since the linear form is nonzero, it is surjective and for all $a \in \mathbb{F}_q$, we have

$$|\{\boldsymbol{c} \in \mathscr{C} \mid \langle \boldsymbol{c}, \boldsymbol{u}\rangle = a\}| = |\ker \langle\,\cdot\,, \boldsymbol{u}\rangle\,| = q^{k-1}.$$

Consequently,

$$\sum_{\boldsymbol{c}\in\mathscr{C}} \chi_1(\langle \boldsymbol{c}, \boldsymbol{u}\rangle) = q^{k-1}\sum_{a\in\mathbb{F}_q}\chi_1(a) = 0,$$

where the last equality is a direct consequence of Lemma 5.12(ii) (since $1 \neq 0$). □

Proof of Theorem 5.8

We have

$$P_{\mathscr{C}^\perp}(x,y) = \sum_{\boldsymbol{c}\in\mathscr{C}^\perp} x^{\mathrm{w_H}(\boldsymbol{c})} y^{n-\mathrm{w_H}(\boldsymbol{c})} = y^n \sum_{\boldsymbol{c}\in\mathscr{C}^\perp} (xy^{-1})^{\mathrm{w_H}(\boldsymbol{c})}.$$

Using Lemma 5.13, we get

$$P_{\mathscr{C}^\perp}(x,y) = y^n \sum_{\boldsymbol{c}\in\mathbb{F}_q^n} \frac{1}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} \chi_1(\langle \boldsymbol{c},\boldsymbol{x}\rangle)(xy^{-1})^{\mathrm{w_H}(\boldsymbol{c})}.$$

For all $a \in \mathbb{F}_q$, set

$$\mathrm{w_H}(a) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if} \quad a \neq 0 \\ 0 & \text{else.} \end{cases}$$

Then, after swapping the sums,

$$\begin{aligned}
P_{\mathscr{C}^\perp}(x,y) &= \frac{y^n}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} \sum_{\boldsymbol{c}\in\mathbb{F}_q^n} \zeta_p^{\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(\langle \boldsymbol{c},\boldsymbol{x}\rangle)} (xy^{-1})^{\mathrm{w_H}(\boldsymbol{c})}. \\
&= \frac{y^n}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} \sum_{\boldsymbol{c}\in\mathbb{F}_q^n} \zeta_p^{\mathrm{Tr}(c_1x_1)+\cdots+\mathrm{Tr}(c_nx_n)} (xy^{-1})^{\mathrm{w_H}(c_1)+\cdots+\mathrm{w_H}(c_n)} \\
&= \frac{y^n}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} \prod_{i=1}^{n} \left(\sum_{c_i\in\mathbb{F}_q} \zeta_p^{\mathrm{Tr}(c_ix_i)} (xy^{-1})^{\mathrm{w_H}(c_i)} \right) \\
&= \frac{y^n}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} \prod_{i=1}^{n} \left(\sum_{c\in\mathbb{F}_q} \chi_c(x_i) (xy^{-1})^{\mathrm{w_H}(c)} \right) \\
&= \frac{y^n}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} \prod_{i=1}^{n} \left(1 + xy^{-1} \sum_{c\in\mathbb{F}_q^\times} \chi_c(x_i) \right).
\end{aligned}$$

Remind that, from Lemma 5.12(i),

$$\sum_{c\in\mathbb{F}_q^\times} \chi_c(x_i) = \begin{cases} q-1 & \text{if} \quad x_i = 0 \\ -1 & \text{else} \end{cases}$$

Therefore,

$$\begin{aligned}
P_{\mathscr{C}^\perp}(x,y) &= \frac{y^n}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} (1+(q-1)xy^{-1})^{n-\mathrm{w_H}(\boldsymbol{x})} (1-xy^{-1})^{\mathrm{w_H}(\boldsymbol{x})} \\
&= \frac{1}{|\mathscr{C}|} \sum_{\boldsymbol{x}\in\mathscr{C}} (y-x)^{\mathrm{w_H}(\boldsymbol{x})} (y+(q-1)x)^{n-\mathrm{w_H}(\boldsymbol{x})}.
\end{aligned}$$

This concludes the proof.

Chapter 6

Reed Solomon codes

Reed–Solomon codes is a family of codes with many remarkable properties. Among others, they are MDS, i.e. their minimum distance equals $n-k+1$ and there exist efficient decoding algorithms to correct any pattern of $\lfloor \frac{d-1}{2} \rfloor$ errors. A major breaktrough in the topic of algebraic codes is due to Sudan [Sud97] who gave in 1997 a polynomial time algorithm allowing to correct beyond the radius $\lfloor \frac{d-1}{2} \rfloor$.

The family of Reed–Solomon codes is of central interest in coding theory since many algebraic constructions of codes such as BCH codes, Goppa codes, alternant codes derive from the family of Reed Solomon codes or generalised Reed Solomon codes.

Reed Solomon codes are practically used for instance in compact discs, DVD's, BluRay's, ADSL, QR codes etc...

6.1 Definition and first properties

Notation 6.1. Let s be a positive integer, we denote respectively by $\mathbb{F}_q[X]_{<s}$ and $\mathbb{F}_q[X]_{\leqslant s}$ the spaces of polynomials of degree less than (resp. less than or equal to) s.

Definition 6.1. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be an n–tuple of **pairwise distinct** elements of $\mathbb{F}_q$ (in particular $n \leqslant q$) and let $k \leqslant n$. The code $\mathbf{RS}_k(\boldsymbol{x})$ is defined as

$$\mathbf{RS}_k(\boldsymbol{x}) \stackrel{\text{def}}{=} \{(f(x_1), \ldots, f(x_n)) \mid f \in \mathbb{F}_q[x]_{<k}\}\,.$$

Remark 27. The code $\mathbf{RS}_k(\boldsymbol{x})$ has a generator matrix of the form

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{k-1} & x_2^{k-1} & \cdots & x_n^{k-1} \end{pmatrix}.$$

which is a truncated Van Der Monde matrix (the $n-k$ last rows are removed).

Proposition 6.2. *Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be an n–tuple of* **pairwise distinct** *elements of $\mathbb{F}_q$ (in particular $n \leqslant q$) and let $k \leqslant n$. The code $\mathbf{RS}_k(\boldsymbol{x})$ has parameters $[n, k, n-k+1]$. It is an MDS code.*

Proof. Consider the map

$$\phi_{k,\boldsymbol{x}} : \left\{ \begin{array}{ccc} \mathbb{F}_q[x]_{<k} & \longrightarrow & \mathbb{F}_q^n \\ f & \longmapsto & (f(x_1), \ldots, f(x_n)). \end{array} \right.$$

This map is injective, indeed, let $f \in \mathbb{F}_q[x]_{<k}$ such that $f(x_1) = \cdots = f(x_n) = 0$. Then f has n distinct roots, while its degree is $< k$ and hence $< n$. Thus $f = 0$. Consequently,

$$\dim_{\mathbb{F}_q} \mathbf{RS}_k(\boldsymbol{x}) = \dim_{\mathbb{F}_q} \mathbb{F}_q[x]_{<k} = k.$$

Next, let $\boldsymbol{c} = (f(x_1), \ldots, f(x_n)) \in \mathbf{RS}_k(\boldsymbol{x})$ be a nonzero codeword. That is $f \neq 0$. Then f has at most $k-1$ distinct roots and hence $\mathrm{w_H}(\boldsymbol{c}) \geqslant n-k+1$. Therefore the minimum distance of $\mathbf{RS}_k(\boldsymbol{x})$ is bounded below by $n-k+1$. It is also bounded above by the same quantity because of the Singleton bound. Thus

$$d_{\min}(\mathbf{RS}_k(\boldsymbol{x})) = n-k+1.$$

□

Remark 28. A major drawback of Reed-Solomon codes is that their length is upper bounded by the size of the alphabet $\mathbb{F}_q$. In particular, compared to Chapter 4 in which we considered the asymptotic behaviour of sequence of codes over a fixed base field whose length was tending to infinity, no such family of RS codes exists since if we consider codes over a fixed base field $\mathbb{F}_q$ all the RS codes have length $\leqslant q$.

Reed Solomon codes form a subfamily of a larger family of codes:

Definition 6.3 (Generalised Reed Solomon codes)**.** Let $x_1, \ldots, x_n$ be pairwise distinct elements of $\mathbb{F}_q$ and $y_1, \ldots, y_n$ be elements of $\mathbb{F}_q^\times$. Then we define the code

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid f \in \mathbb{F}_q[x]_{<k}\}.$$

Notice that since the y_i's are nonzero, the map

$$\left\{ \begin{array}{ccc} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ (u_1, \ldots, u_n) & \longmapsto & (y_1 u_1, \ldots, y_n u_n) \end{array} \right.$$

is an isomorphism that preserves the Hamming weights (i.e. an isometry). Since this map sends $\mathbf{RS}_k(\boldsymbol{x})$ onto $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$, then both codes are isometric and hence have the same dimension and minimum distance. In particular, a GRS code is MDS too.

6.2 Duality

6.2.1 The special case of full support codes

Theorem 6.4. *Let $x_1, \ldots, x_n \in \mathbb{F}_q$ be pairwise distinct and such that $n = q$, i.e. the set $\{x_1, \ldots, x_n\}$ equals the set of elements of $\mathbb{F}_q$. Then for all $k \leqslant n$,*

$$\mathbf{RS}_k(\boldsymbol{x})^\perp = \mathbf{RS}_{n-k}(\boldsymbol{x}).$$

To prove this result we need the following lemma.

Lemma 6.5. *For all $0 < t \leqslant q-1$, we have*

$$\sum_{\alpha \in \mathbb{F}_q} \alpha^t = \begin{cases} 0 & \text{if} \quad t < q-1 \\ -1 & \text{if} \quad t = q-1. \end{cases}$$

Proof. Remind that the multiplicative group $\mathbb{F}_q^\times$ is cyclic and hence there is an element $\gamma \in \mathbb{F}_q^\times$ such that $\{1, \gamma, \gamma^2, \ldots, \gamma^{q-2}\} = \mathbb{F}_q^\times$. Now,

$$\sum_{\alpha \in \mathbb{F}_q^\times} \alpha^t = \sum_{i=0}^{q-1} \gamma^{it}.$$

If $t \neq q-1$, then $\gamma^t \neq 1$ and we have the sum of the elements of a geometric progression:

$$\sum_{\alpha \in \mathbb{F}_q^\times} \alpha^t = \frac{1 - \gamma^{t(q-1)}}{1 - \gamma^t}$$

which is equal to 0 since $\gamma^{q-1} = 1$. On, the other hand if $t = q-1$, then we have

$$\sum_{\alpha \in \mathbb{F}_q^\times} \alpha^{q-1} = \sum_{\alpha \in \mathbb{F}_q^\times} 1 = q-1$$

which equals 1 in $\mathbb{F}_q$. This concludes the proof. □

Proof of Theorem 6.4. As noticed in Remark 27, the codes $\mathbf{RS}_k(\boldsymbol{x})$ and $\mathbf{RS}_{n-k}(\boldsymbol{x})$ have generator matrices respectively of the form

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{k-1} & x_2^{k-1} & \cdots & x_n^{k-1} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{n-k+1} & x_2^{n-k+1} & \cdots & x_n^{n-k+1} \end{pmatrix}.$$

We will prove that any row of the first matrix is orthogonal to any row of the second. First notice that the first row of the left hand matrix is orthogonal to the first row of the right hand matrix since the product is

$$\langle (1,\dots,1),(1,\dots,1)\rangle = \sum_{i=1}^{n} 1 = n$$

and since $n = q$ the sum is zero in $\mathbb{F}_q$. Next, for all $0 \leqslant i < k$ and all $0 \leqslant j < n-k$ such that $i + j \neq 0$, we have[1]

$$\left\langle (x_1^i,\dots,x_n^i),(x_i^j,\dots,x_n^j)\right\rangle = \sum_{\ell=0}^{n} x_\ell^{i+j} = \sum_{x\in\mathbb{F}_q} x^{i+j}$$

which is zero from Lemma 6.5. By linearity, we deduce that every codeword of $\mathbf{RS}_k(\boldsymbol{x})$ is orthogonal to every codeword of $\mathbf{RS}_{n-k}(\boldsymbol{x})$ and hence,

$$\mathbf{RS}_{n-k}(\boldsymbol{x}) \subseteq \mathbf{RS}_k(\boldsymbol{x})^{\perp}.$$

The converse inclusion is obtained by noting that, thanks to Propositions 6.2 and 5.3, both codes have the same dimension □

The general case

In general, the dual of a Reed–Solomon code is not a Reed–Solomon code but it is always a *generalised* Reed–Solomon code. More precisely we always have the following result.

Theorem 6.6. *Let $\boldsymbol{x} = (x_1,\dots,x_n) \in \mathbb{F}_q^n$ where the x_i's are pairwise distinct and $\boldsymbol{y} = (y_1,\dots,y_n) \in (\mathbb{F}_q^\times)^n$. Then,*

$$\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})^{\perp} = \mathbf{GRS}_{n-k}(\boldsymbol{x},\boldsymbol{y}'),$$

where $\boldsymbol{y}' = (y_1',\dots,y_n')$ with for all i,

$$y_i' \stackrel{def}{=} -\frac{1}{y_i \prod_{\substack{j=1\\ j\neq i}}^{n}(x_j - x_i)}.$$

To prove the Theorem, we first use the following lemma.

Lemma 6.7. $\prod_{\alpha\in\mathbb{F}_q^\times} \alpha = -1$.

Proof. The elements of $\mathbb{F}_q^\times$ are the roots of the polynomial $X^{q-1}-1$. By the formulas relating coefficients and elementary symmetric functions on the roots,

$$\prod_{\alpha\in\mathbb{F}_q^\times} = (-1).(-1)^{q-1}.$$

This quantity is -1 if q is odd and 1 if q is even, but if q is even, then $1 = -1$. □

[1] With the convention $0^0 = 1$.

Proof of Theorem 6.6. First consider the case of a full support i.e. where $n = q$ and hence $\{x_1, \ldots, x_n\} = \mathbb{F}_q$. In this situation, for all $i \in \{1, \ldots, n\}$,

$$\prod_{j \neq i}(x_j - x_i) = \prod_{\alpha \in \mathbb{F}_q^\times} \alpha = -1,$$

where the lat equality is due to Lemma 6.7. Therefore in the case of a full support,

$$\forall i \in \{1, \ldots, n\}, \quad y_i' = -\frac{1}{y_i}\cdot$$

Then, let $f \in \mathbb{F}_q[X]_{<k}$ and $g \in \mathbb{F}_q[X]_{<n-k}$, then

$$\begin{aligned}\langle (y_1 f(x_1), \ldots, y_n f(x_n)), (y_1' g(x_1), \ldots, -y_n' g(x_n)) \rangle &= \sum_{i=1}^{n} y_i y_i' f(x_i) g(x_i) \\ &= -\langle (f(x_1), \ldots, f(x_n)), (g(x_1), \ldots, g(x_n)) \rangle .\end{aligned}$$

Since the words $(f(x_1), \ldots, f(x_n))$ and $(g(x_1), \ldots, g(x_n))$ are respectively in $\mathbf{RS}_k(\boldsymbol{x})$ and $\mathbf{RS}_{n-k}(\boldsymbol{x})$ their inner product is 0 thanks to Theorem 6.4. Therefore

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \subseteq \mathbf{GRS}_{n-k}(\boldsymbol{x}, \boldsymbol{y})^\perp$$

and the converse inclusion holds due to the equality of dimension of the codes.

In the general case, first set

$$Q(X) \stackrel{\text{def}}{=} \prod_{\alpha \in \mathbb{F}_q \setminus \{x_1, \ldots, x_n\}} (X - \alpha).$$

Next, notice that for all $j \in \{1, \ldots, n\}$

$$\begin{aligned} y_j' &= -\frac{1}{y_j \prod_{\substack{i=1 \\ i \neq j}}^{n} (x_j - x_i)} \\ &= \frac{\prod_{\alpha \in \mathbb{F}_q \setminus \{x_1, \ldots, x_n\}} (x_j - \alpha)}{y_j \prod_{\alpha \in \mathbb{F}_q \setminus \{j\}} (x_j - \alpha)} \\ &= \frac{\prod_{\alpha \in \mathbb{F}_q \setminus \{x_1, \ldots, x_n\}} (x_j - \alpha)}{y_j \prod_{\alpha \in \mathbb{F}_q^\times} \alpha} \\ &= -\frac{1}{y_j} Q(x_j). \end{aligned}$$

Consequently, let $f \in \mathbb{F}_q[X]_{<k}$ and $g \in \mathbb{F}_q[X]_{<n-k}$,

$$\langle (y_1 f(x_1), \ldots, y_n f(x_n)), (y_1'(x_1), \ldots, y_n' g(x_n)) \rangle = -\sum_{i=1}^{n} Q(x_i) f(x_i) g(x_i)$$

Since Q vanishes at every $\alpha \in \mathbb{F}_q \setminus \{x_1, \ldots, x_n\}$,

$$\langle (y_1 f(x_1), \ldots, y_n f(x_n)), (y'_1(x_1), \ldots, y'_n g(x_n)) \rangle = -\sum_{\alpha \in \mathbb{F}_q} Q(\alpha) f(\alpha) g(\alpha). \tag{6.1}$$

Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_q)$ be an n–tuple of pairwise distinct elements in $\mathbb{F}_q$, i.e. such that $\{\alpha_1, \ldots, \alpha_q\} = \mathbb{F}_q$. Then, the right hand side of (6.1) is the inner product in $\mathbb{F}_q^q$ of the words $(Q(\alpha_1)f(\alpha_1), \ldots, Q(\alpha_q)f(\alpha_q))$ and $(g(\alpha_1), \ldots, g(\alpha_q))$ which are respectively in $\mathbf{RS}_{k+q-n}(\boldsymbol{\alpha})$ and $\mathbf{RS}_{n-k}(\boldsymbol{\alpha})$ which are dual to each other thanks to Theorem 6.4. Consequently,

$$\langle (y_1 f(x_1), \ldots, y_n f(x_n)), (y'_1(x_1), \ldots, y'_n g(x_n)) \rangle = 0$$

and hence $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \subseteq \mathbf{GRS}_{n-k}(\boldsymbol{x}, \boldsymbol{y}')^\perp$ and the converse inclusion is due to the equality of the dimensions of these codes. □

6.3 Alternant codes

Actually, most of the algebraic code constructions derive from Reed Solomon codes. BCH codes which will be studied in Chapter 9 can be constructed from Reed Solomon codes. A larger family of codes derived from Reed Solomon of the family of alternant codes which contains several subfamilies such as BCH codes and classical Goppa codes. Alternant codes are nothing but subfield subcodes of Reed Solomon codes. We refer to § 1.4.3 for the definition and properties of subfield subcodes.

A motivation for such a construction is that Reed Solomon codes have optimal parameters and benefit from efficient polynomial time decoding algorithms (see Chapter 8). On the other hand their major drawback, is that their length should be less than or equal to the size of the alphabet. Alternant codes have not as good parameters as Reed Solomon codes but can be defined over arbitrary small fields, even over $\mathbb{F}_2$.

Definition 6.8. Let $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ be a support, i.e. a vector of pairwise distinct entries. Let $\boldsymbol{y} \in (\mathbb{F}_{q^m}^\times)^n$ and r be an integer. The code $\mathcal{A}_r(\boldsymbol{x}, \boldsymbol{y})$ is defined as

$$\mathcal{A}_r(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \left(\mathbf{GRS}_r(\boldsymbol{x}, \boldsymbol{y})^\perp\right) \cap \mathbb{F}_q^n$$

Remark 29. Note that, since the dual of a generalised Reed Solomon code (see Theorem 6.6) is another generalised Reed Solomon code, an altenant code is nothing but a subfield subcode of a generalised Reed Solomon code. The choice of a definition involving a dual is due to convenience. For instance the parameters are much easier to define from this definition as you can see in the following statement.

Proposition 6.9. *The code $\mathcal{A}_r(\boldsymbol{x}, \boldsymbol{y})$ has parameters $[n, \geqslant n - mr, \geqslant r + 1]$.*

Proof. From Theorem 6.6, the code $\mathbf{GRS}_r(\boldsymbol{x}, \boldsymbol{y})^\perp$ is a generalised Reed Solomon code of dimension $n - r$ and hence of parameters $[n, n - r, r + 1]$. Then we conclude using Proposition 1.26. □

Comment on Proposition 6.9 Note that Proposition 6.9, only gives lower bounds for the dimension and the minimum distance. It is natural to wonder if these bounds are sharp or not. Actually, the lower bound on the dimension is sharp in general. Some particular constructions of altenant codes such as the so–called classical Goppa codes (see [MS77, Chapter 12,§ 3]) have a dimension which exceeds the lower bound but this property is rather rare.

On the other hand, the lower bound on the minimum distance is far from being sharp in general. In particular, combinatorial methods permit to prove that some sequence of alternant codes of constant rate reach the Gilbert Varshamov bound (see [HP03, Theorem 13.5.1]). However, since the actual minimum distance is hard to determine the lower bound given from Proposition 6.9 is informative even if it may be far below the actual minimum distance. Moreover, this lower bound gives a decoding radius: if we are able to decode a Reed Solomon up to some radius t then applying the same decoder to the subfield subcode, we are able to decode the subfield subcode up to the same radius. In chapter 8 we will show that the unambiguous decoding, of Reed–Solomon can be performed in polynomial time. From this decoding algorithm, one deduces a decoding algorithm for an alternant code $\mathcal{A}_r(\boldsymbol{x}, \boldsymbol{y})$ which corrects up to $\lfloor \frac{r}{2} \rfloor$ errors.

Chapter 7

MDS codes

Maximum distance separable codes are an extremely exciting object of study: they are exceptional combinatorial objects and have various applications for instance in symmetric cryptography, since some of their generator matrices provide excellent primitive in the diffusion step of block ciphers.

In the previous chapter we saw an explicit example of maximum distance separable codes which are Reed–Solomon codes. It is well known that many MDS codes are not generalised Reed Solomon codes. On the other hand the complete classification of MDS codes is far from being known and many questions on MDS codes still remain open. A major and still open question is:

Question 7.1. What is the maximum length of a non trivial MDS code over $\mathbb{F}_q$?

We explain below what we mean by *non trivial*.

7.1 MDS codes, definition and first examples

Generator and parity check matrices of MDS codes have very particular properties which are explained in the following statement.

Definition 7.1. An $[n, k, d]_q$ MDS code is a code reaching Singleton bound. That is to say a code such that $d = n - k + 1$.

Example 7.2. The first and most elementary examples are:

- The *full code* : $\mathbb{F}_q^n$ i.e. the whole ambient space is $[n, n, 1]_q$ and hence is MDS;
- By convention, we consider that the zero code $\{0\}$ has minimum distance $n+1$. Hence this code is MDS too. The convention is relevant since, there is no nonzero codeword in the zero code, hence its minimum distance should be larger than n. On the other hand, the choice $d = n + 1$ make the code satisfy Singleton bound. Note also that it will be proved further in Theorem 7.6, that the dual of an MDS code is MDS and $\{0\}$ is the dual of the full code which is MDS.

- The repetition code spanned by $(1\ 1\ \cdots\ 1)$ is $[n, 1, n]_q$ and hence is MDS.
- The repetition code defined in § 1.3.1 is $[n, n-1, 2]_q$. It is dual to the repetition code and is MDS too.

Example 7.3. As explained in Chapter 6, generalised Reed Solomon codes are MDS.

7.2 Some properties of MDS codes

7.2.1 Generator and parity–check matrices of MDS codes

Proposition 7.4. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a of dimension k. Let $\boldsymbol{G} \in \mathfrak{M}_{k,n}(\mathbb{F}_q)$ and $\boldsymbol{H} \in \mathfrak{M}_{n-k,n}(\mathbb{F}_q)$ be respective full rank generator and parity–check matrices of $\mathscr{C}$. Then $\mathscr{C}$ is MDS if and only if one of these two assertions is satisfied.*

(1) any $k \times k$ minor of $\boldsymbol{G}$ is nonzero;

(2) any $(n-k) \times (n-k)$ minor of $\boldsymbol{H}$ is nonzero.

Proof. The proof of (1) is in the same spirit as the alternative proof for the Singleton bound (see Remark 23). Assume that some $k \times k$ minor of $\boldsymbol{G}$ corresponding to the columns $i_1, \ldots, i_k$ is zero. By Gaussian elimination, one can construct a nonzero linear combination $\boldsymbol{c} \in \mathscr{C}$ of the rows of $\boldsymbol{G}$ such that $c_{i_1} = \cdots = c_{i_k} = 0$. Since $\boldsymbol{G}$ has full rank and $\boldsymbol{c}$ has be constructed as a nonzero linear combination of the rows of $\boldsymbol{G}$ and hence is nonzero. Thus the Hamming weight of $\boldsymbol{c}$ is less than or equal to $n-k$, and hence $\mathscr{C}$ is not MDS.

Conversely, if any $k \times k$ minor is nonzero, then one cannot construct a codeword of weight $\leqslant n-k$. This proves (1).

For (2), remind that Proposition 1.11 asserts that the minimum distance of $\mathscr{C}$ is the least number of linearly linked columns of $\boldsymbol{H}$. Since $\boldsymbol{H}$ is $(n-k) \times n$ any $(n-k+1)$–tuple of columns is linked and being MDS is equivalent to the fact that any $(n-k)$–tuple of columns is independent and hence that any $(n-k) \times (n-k)$ minor is nonzero. □

As a consequence, given a code $\mathscr{C}$ and a generator matrix $\boldsymbol{G} = (I_k \mid A)$ in systematic form, the right–hand block A has a very particular property.

Proposition 7.5. *Let $\mathscr{C}$ be an MDS code and $\boldsymbol{G} = (I_k \mid A)$ be its systematic generator matrix. Then,* **any** *minor of A is nonzero. In particular, any entry of A is nonzero.*

Proof. By Proposition 7.4 and since $k \times k$ of A are nothing but $k \times k$ minors of $\boldsymbol{G}$, the result is clear for $k \times k$ minors. Let us consider $a \times a$ minors for $a < k$. Consider an $a \times a$ minor of A (i.e. of $\boldsymbol{G}$) which corresponds to the rows $R_{i_1}, \ldots, R_{i_a}$ of $\boldsymbol{G}$ and the columns $C_{j_1}, \ldots, C_{j_a}$ such that for all $1 \leqslant i \leqslant a$, we have $j_i > k$. Denote by M the corresponding submatrix of A so that $\det M$ is the minor we are studying. Let $i'_1, \ldots, i'_{k-a}$ be a sequence of indexes such that

$$\{i_1, \ldots, i_a\} \cup \{i'_1, \ldots, i'_{k-a}\} = \{1, \ldots, k\}$$

and consider the minor corresponding to the rows $i'_1, \ldots, i'_{k-a}, i_1, \ldots, i_a$ and the columns $i'_1, \ldots, i'_{k-a}, j_1, \ldots, j_a$. As a submatrix, it is of the form

$$\begin{pmatrix} I_{n-a} & (*) \\ (0) & M \end{pmatrix}$$

Therefore, the determinant of the matrix is a $k \times k$ minor of $\boldsymbol{G}$ which is nonzero by assumption. Moreover, this determinant is nothing but that of M, hence the corresponding $a \times a$ minor of A is nonzero. This concludes the proof. □

For an MDS code $\mathscr{C}$ of length n for n even and dimension $n/2$, the matrix A of Proposition 7.5 is a square matrix and such matrix has only nonzero minors. Such matrices are said to be *MDS* and are of central interest in symmetric cryptography since they have excellent diffusion properties.

7.2.2 Duality

Theorem 7.6. *The dual of an MDS code is MDS.*

Proof. From Proposition 5.3, a generator matrix of a code is a parity–check matrix of its dual. Thus, the result is a direct consequence of Proposition 7.4. □

7.2.3 Puncturing and shortening MDS codes

Proposition 7.7. *Let $\mathscr{C} \in \mathbb{F}_q^n$ be an MDS code of dimension k. Let $I \subseteq \{1, \ldots, n\}$. Then, the punctured and shorten codes $\mathcal{P}_I(\mathscr{C})$ and $\mathcal{S}_I(\mathscr{C})$ are MDS and have respective parameters $[n - |I|, k, n - k - |I| + 1]_q$ and $[n - |I|, k - |I|, n - k + 1]_q$.*

Proof. Let $\boldsymbol{G}$ be a full–rank generator matrix of $\mathscr{C}$. From Proposition 7.4(1), any $k \times k$ minor of $\boldsymbol{G}$ is nonzero. Now let $\boldsymbol{G}'$ be the matrix obtained from $\boldsymbol{G}$ by removing the columns with index in I. From Lemma 1.19, $\boldsymbol{G}'$ is a generator matrix for $\mathcal{P}_I(\mathscr{C})$. Next, obviously any $k \times k$ minor of $\boldsymbol{G}'$ is nonzero, thus $\mathcal{P}_I(\mathscr{C})$ is MDS. Moreover, the MDS property entails that $\boldsymbol{G}'$ is full rank and hence $\mathcal{P}_I(\mathscr{C})$ has parameters $[n - |I|, k, n - k - |I| + 1]_q$

Next, from Theorem 5.5, we have $\mathcal{S}_I(\mathscr{C}) = (\mathcal{P}_I(\mathscr{C}^\perp))^\perp$, moreover, from Theorem 7.6, $\mathscr{C}^\perp$ is MDS and we proved above that $\mathcal{P}_I(\mathscr{C}^\perp)$ is MDS. Using Theorem 7.6 again we prove that $(\mathcal{P}_I(\mathscr{C}^\perp))^\perp$ is MDS and hence $\mathcal{S}_I(\mathscr{C})$ is MDS. □

7.3 Length of MDS codes, the MDS conjecture

As explained in Chapter 6, Reed Solomon codes are MDS but their major drawback is that their length is bounded above by the size of their base field. It is actually possible to extend Reed–Solomon to MDS codes of length $q + 1$ and the existence of longer MDS codes is still an open problem. Basically, outside Reed–Solomon codes, the only long MDS code which are known are the trivial MDS codes of Example 7.2 and the MDS codes described by the following statement which we will admit.

Proposition 7.8 (Exceptional MDS codes in characteristic 2 (admitted))**.** *Let $q = 2^m$ for some $m \geqslant 1$. There exists an MDS code of length $q+2$ and dimension 3 and an MDS code of length $q+2$ and dimension $q-1$. The two codes are dual to each other.*

Example 7.9. Consider the finite field $\mathbb{F}_4$ defined as $\mathbb{F}_2[\alpha]$ with α such that $\alpha^2+\alpha+1=0$. Over $\mathbb{F}_4$, the code with generator matrix

$$\begin{pmatrix} 0 & 1 & \alpha & \alpha+1 & 0 & 1 \\ 0 & 1 & \alpha+1 & \alpha & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

is MDS.

Remark 30. The exceptional MDS codes come from very particular combinatorial and geometric properties in characteristic 2. They can be constructed from wonderful and totally counter–intuitive geometric objects called *hyperovals* which are subset of the projective plane which exist only in characteristic 2.

Now we are able to state the well–known MDS conjecture.

Conjecture 1 (MDS conjecture)**.** *Any MDS code over $\mathbb{F}_q$ has length less than or equal to $q+1$ but*

- *the trivial MDS codes of Example 7.2;*
- *the exceptional MDS codes in characteristic 2 of Proposition 7.8.*

What do we know about this conjecture The first result one can prove on the length of MDS codes is:

Theorem 7.10. *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a non trivial (i.e. different from the codes of Example 7.2) MDS code of length n dimension k. Then*

$$n \leqslant q+k-1.$$

In particular, the conjecture is true for $k=2$.

Proof. Let us prove the result for $k=2$. Let $\mathscr{C}$ be an MDS code of dimension 2 and $\boldsymbol{G} \in \mathfrak{M}_{2,n}(\mathbb{F}_q)$ be a full–rank generator matrix of $\mathscr{C}$. Since $\mathscr{C}$ is MDS, any pair of columns of $\boldsymbol{G}$ should be non collinear. Therefore, if we consider for any column C_i of $\boldsymbol{G}$ the vector line $L_i \subseteq \mathbb{F}_q^2$ spanned by C_i, the lines L_i should be pairwise distinct.

On the other hand it is well–known that in the plane $\mathbb{F}_q^2$ there are exactly $q+1$ lines[1]:

- the line of equation $y=0$
- the lines of equation $y=\alpha x$ for $\alpha \in \mathbb{F}_s$

[1]If you like projective geometry it is nothing but the number of elements of the projective line $\mathbb{P}^1(\mathbb{F}_q)$

Thus, we get $n \leqslant q+1$ and the theorem is proved for $k = 2$.

Now, if $k \geqslant 2$, shorten the code $\mathscr{C}$ at $k-2$ positions. From Proposition 7.7, the shortened code is MDS of parameters $[n-(k-2), 2, n-k+1]_q$. Thus, since the result has been proved for MDS codes of dimension 2 we can apply it to the shortened code, which yields

$$n - (k-2) \leqslant q+1 \qquad \Longrightarrow \qquad n \leqslant q+k-1.$$

□

Corollary 7.11. *Let $\mathscr{C}$ be a non trivial MDS code of length n and dimension k, then $k \leqslant q-1$.*

Proof. Applying Theorem 7.10 to $\mathscr{C}^\perp$ we get

$$n \leqslant q + (n-k) - 1 \qquad \Longrightarrow \qquad k \leqslant q-1.$$

□

Finally a recent breaktrough due to Simeon Ball and Jan De Beule provides a partial proof of the conjecture.

Theorem 7.12 ([BDB12])**.** *Let $q = p^m$ and $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a nontrivial MDS code of dimension k different from the exceptional codes of Proposition 7.8. If $k \leqslant 2p-2$, then $n \leqslant q+1$. In particular, over a prime field $\mathbb{F}_p$ the conjecture is true.*

Chapter 8

Decoding (generalised) Reed Solomon codes

Another wonderful feature of generalised Reed Solomon codes is that they benefit to efficient decoding algorithms. In this chapter, we first present Berlekamp Welch algorithm which permits to correct errors up to half the minimum distance. In the end of the 90's two consecutive breakthroughs due to Sudan [Sud97] and Guruswami Sudan [GS99] permitted first to correct errors beyond half the minimum distance and then to correct errors up to the so called *Johnson bound* (see §8.2.2). The price to pay when correcting errors beyond half the minimum distance is that instead of solving a bounded decoding problem, the algorithm solves the *list decoding problem* (see § 2.1.1).

First notice that, as soon as we have a decoding algorithm for Reed–Solomon codes, then we have a decoding algorithm for generalised Reed–Solomon codes. Indeed, assume that we have a decoding algorithm for a Reed Solomon code $\mathbf{RS}_k(\boldsymbol{x})$ correcting up to t errors.

Then, let $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ and assume we received $\boldsymbol{v} = \boldsymbol{c} + \boldsymbol{e} = (v_1, \ldots, v_n)$ where $\boldsymbol{e} \in \mathbb{F}_q^n$ with $\mathrm{w_H}(\boldsymbol{e}) \leqslant t$. Then, it suffices to notice that $(y_1^{-1}c_1, \ldots, y_n^{-1}c_n)$ is an element of $\mathbf{RS}_k(\boldsymbol{x})$. Therefore, compute $(y_1^{-1}v_1, \ldots, y_n^{-1}v_n)$ and apply our algorithm for decoding $\mathbf{RS}_k(\boldsymbol{x})$. This algorithm outputs a word $\boldsymbol{c}' = (c'_1, \ldots, c'_n)$ such that $(y_1c'_1, \ldots, y_nc'_n) = (c_1, \ldots, c_n)$.

Thus, from now on, we only consider Reed–Solomon codes. According to the previous remarks, it does not represents any loss of generality.

8.1 Unique decoding : Berlekamp Welch algorithm

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be an n–tuple of pairwise distinct elements of $\mathbb{F}_q$ and consider the code $\mathbf{RS}_k(\boldsymbol{x})$ with $1 \leqslant k < n$. Set

$$t \stackrel{\text{def}}{=} \left\lfloor \frac{n-k}{2} \right\rfloor .$$

Let $\boldsymbol{c} = (f(x_1), \ldots, f(x_n)) \in \mathbf{RS}_k(\boldsymbol{x})$ with $f \in \mathbb{F}_q[X]_{<k}$. Let $\boldsymbol{e} \in \mathbb{F}_q^n$ be *the error*: $\mathrm{w_H}(\boldsymbol{e}) \leqslant t$. Set

$$\boldsymbol{y} \stackrel{\text{def}}{=} \boldsymbol{c} + \boldsymbol{e}$$

the *received word*. Basically $\boldsymbol{y}$ is known while $\boldsymbol{c}, \boldsymbol{e}$ are not. We introduce the following (unknown) polynomial

$$E(X) \stackrel{\text{def}}{=} \prod_{i \ \mid \ e_i \neq 0} (X - x_i).$$

The key of the algorithm reposes on the following identity:

$$\forall i \in \{1, \ldots, n\}, \quad y_i E(x_i) = f(x_i) E(x_i). \tag{8.1}$$

Indeed, either $e_i \neq 0$ but in that case, by the very definition of E, we have $E(x_i) = 0$ or $e_i = 0$ in which case, $y_i = f(x_i)$.

Remind that y_i's and x_i's are known, hence the unknowns of the system of equations given by (8.1) are the coefficients of E and those of f. Unfortunately, this system is non linear because of the term $f(x_i)E(x_i)$. For this reason, we proceed to a linearisation. Set

$$N \stackrel{\text{def}}{=} Ef.$$

Then, (8.1) becomes

$$\forall i \in \{1, \ldots, n\}, \quad y_i E(x_i) = N(x_i). \tag{8.2}$$

Here, (8.2) provides a new system of n equations whose unknowns are the coefficients of E and N. The number of unknowns is $k + 2t + 1$ since $E \in \mathbb{F}_q[X]_{\leqslant t}$ and hence has $t + 1$ coefficients and $N \in \mathbb{F}_q[X]_{\leqslant k-1+t}$ and hence has $k + t$ coefficients. This new system is linear and has a nontrivial solution given by the pair (E, Ef). Moreover, any other nontrivial solution allows to find f, thanks to the following result.

Theorem 8.1. *Let (E_1, N_1) and $(E_2, N_2) \in \mathbb{F}_q[X]_{\leqslant \lfloor \frac{n-k}{2} \rfloor} \times \mathbb{F}_q[X]_{<k+\lceil \frac{n-k}{2} \rceil}$ be two pairs of nonzero solutions of (8.2). Then $E_1, E_2 \neq 0$ and*

$$\frac{N_1}{E_1} = \frac{N_2}{E_2} = f.$$

Proof. If $E_1 = 0$, then from (8.2) the polynomial N_1 has n distinct roots while its degree is $< k + \lceil \frac{n-k}{2} \rceil \leqslant n$. Thus $N_1 = 0$ which contradicts the fact that the pair (E_1, N_1) is nonzero.

Now, set $R \stackrel{\text{def}}{=} N_1 E_2 - N_2 E_1$. We have

$$\deg(R) \leqslant k + \left\lfloor \frac{n-k}{2} \right\rfloor + \left\lceil \frac{n-k}{2} \right\rceil - 1 \leqslant n - 1.$$

On the other hand, using the fact that $(E_1, N_1), (E_2, N_2)$ are solutions of (8.2),

$$\begin{aligned} \forall i \in \{1, \ldots, n\}, \quad R(x_i) &= N_1(x_i)E_2(x_i) - N_2(x_i)E_1(x_i) \\ &= y_i E_1(x_i)E_2(x_i) - y_i E_1(x_i)E_2(x_i) \\ &= 0. \end{aligned}$$

Therefore, R has n distinct roots while its degree is less than n, hence this polynomial is 0. This proves the equality $\frac{N_1}{E_1} = \frac{N_2}{E_2}$. Hence the fraction $\frac{N}{E}$ is well defined and is the same for every nonzero pair (E, N) solution to (8.2). Since (E, fE) is solution, we get the result. □

Algorithm 6: Berlekamp Welch algorithm

Input : $\boldsymbol{y} \in \mathbb{F}_q^n$
Output: A codeword $\boldsymbol{c} \in \mathbf{RS}_k(\boldsymbol{x})$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ and $\mathrm{w_H}(\boldsymbol{e}) \leqslant t$ if exists. Else returns "?"

Let (E_0, N_0) be a nonzero solution of (8.2);
if $E_0 \nmid N_0$ **then**
  return *"?"*;
else
  Set $f \stackrel{\text{def}}{=} \frac{N_0}{E_0}$;
end
if $\deg f \geqslant k$ *or* $d_H(\boldsymbol{y}, (f(x_1), \ldots, f(x_n))) > t$ **then**
  return *"?"*;
else
  return $(f(x_1), \ldots, f(x_n))$
end

8.1.1 Complexity

By solving a linear system

The most expensive part of the algorithm is the resolution of the linear system (8.2), which has n equations and about n unknowns and hence costs $O(n^3)$ (or $O(n^\omega)$ if you like fast linear algebra). After this resolution, the computation of f is done by performing an Euclidean division of N_0 by E_0 which have respective degrees $\leqslant t$ and $\leqslant t + k$ which costs $O(tk)$ and the evaluation of f at the x_i's which costs $O(kn)$ (by iterating n times the Horner evaluation scheme). Therefore, all the operations after the resolution of the linear system are in $O(n^2)$, which leads to:

Theorem 8.2. *The complexity of Berlekamp Welch algorithm is* $O(n^3)$.

Using extended Euclid algorithm

Actually, linear algebra can be avoided as follows. Let $Y(X) \in \mathbb{F}_q[X]_{<n}$ be the Lagrabge interpolation polynomial of the received vector $\boldsymbol{y}$. That is, Y is the unique polynomial of degree $< n$ satisfying

$$\forall i \in \{1, \ldots, n\}, Y(x_i) = y_i.$$

In addition, set

$$\Pi(X) \stackrel{\text{def}}{=} \prod_{i=1}^{n} (X - x_i).$$

Then, system (8.2) can be reformulated as

$$E(X)Y(X) \equiv N(X) \mod \Pi(X). \tag{8.3}$$

Indeed, by the Chinese remainder theorem, being congruent modulo Π means being congruent modulo $(X - x_i)$ for any $i \in \{1, \ldots, n\}$ and being congruent modulo $X - x_i$ means having the same evaluation at x_i.

Then, (8.3) is equivalent to

$$U(X)\Pi(X) + E(X)Y(X) = N(X)$$

for some polynomial V.

On the beginning, one only knows $Y(X)$ which can be computed from the received word and $\Pi(X)$ which only depends on the x_i's and hence can be pre-computed once for good. Then the idea is to apply Extended Euclid algorithm to the pair (Y, Π) and stop at the good step. To clarify that point, let us recall how extended Euclid algorithm works while using the current notation. We refer the reader to [Dem09, § 1.5.3] for further details.

Algorithm 7: Extended Euclid algorithm

Inputs : $\Pi, Y \in \mathbb{F}_q[X]$

Outputs: $E, V, N \in \mathbb{F}_q[X]$ such that $N = \gcd(\Pi, Y)$ and $U\Pi + EY = N$.

$U_0 = E_1 \leftarrow 1$
$V_1 = E_0 \leftarrow 0$
$N_0 \leftarrow \Pi$
$N_1 \leftarrow Y$
$i \leftarrow 0$
repeat
  Set Q, N_{i+2} the quotient and remainder of the Euclidean division: $N_i = N_{i+1}Q + N_{i+2}$
  $E_{i+2} \leftarrow E_i - E_{i+1}Q$
  $V_{i+2} \leftarrow V_i - V_{i+1}Q$
  $i \leftarrow i + 1$
until $N_{i+1} = 0$;
return (E_i, V_i, N_i)

In extended Euclid Algorithm, a loop invariant is that for all $i \geqslant 0$,

$$U_i\Pi + E_iY = N_i.$$

On the other hand the degrees of E_i, V_i, N_i are loop variants. By definition of Euclidean division, the sequence $(\deg N_i)_i$ is strictly decreasing. Moreover, we have the following lemma.

Lemma 8.3. *For any $i \geqslant 0$, we have*

$$\deg E_i \leqslant \deg E_{i+1}$$
$$\text{and} \quad \deg E_{i+1} = \deg \Pi - \deg N_i.$$

Proof. When the algorithm starts, $N_0 = \Pi$ and $E_0 = 0$, $E_1 = 1$, hence the property is trivially satisfied (using the convention $\deg 0 = -\infty$).

Let us prove by induction that the property holds at any step. First, by definition of the Euclidean division, we have at Step 7:

$$\deg Q = \deg N_{i+1} - \deg N_{i+2}.$$

Next, since by assumption, on the previous step we had $\deg E_i \leqslant \deg E_{i+1}$, then

$$\deg E_{i+2} = \deg(E_i - E_{i+1}Q) = \deg Q + \deg E_{i+1} \geqslant \deg E_{i+1},$$

which, by induction, yields $\deg E_{i+2} = \deg \Pi - \deg N_{i+1}$. This yields the result. □

Now, the idea to solve our decoding problem is to stop the extended Euclid algorithm at a well–chosen intermediary step.

Proposition 8.4. *Let i_0 be the least degree such that $\deg N_{i_0} > k + \lceil \frac{n-k}{2} \rceil - 1$. Then $\deg N_{i_0+1} \leqslant k + \lceil \frac{n-k}{2} \rceil - 1$ and $\deg E_{i_0+1} \leqslant \lfloor \frac{n-k}{2} \rfloor$*

Proof. From Lemma 8.3, we have

$$\begin{aligned}\deg E_{i_0+1} &= \deg \Pi - \deg N_{i_0} \\ &\leqslant n - \left(k + \left\lceil \frac{n-k}{2} \right\rceil\right) \\ &\leqslant \left\lfloor \frac{n-k}{2} \right\rfloor.\end{aligned}$$

□

Therefore, by stopping the extended Euclid algorithm at the i_0–th step and returning (E_{i_0+1}, N_{i_0+1}) yields a valid solution of the system without performing Gaussian elimination. This represents a significant speedup since the naive implementation of extended Euclid algorithm runs in $O(n^2)$ [BCG$^+$17, Thm. 6.1]. In addition, thanks to fast arithmetic, this complexity can be reduced to almost linear time i.e. $O(nP(\log(n)))$ for some polynomial P. See [BCG$^+$17, Thm. 6.16] for further details.

8.2 List decoding

A fundamental result in coding theory is that, out of trivial codes (e.g. $\mathbb{F}_q^n$, repetition codes), Hamming and Golay codes (see [Dem09, Chapter 13.2]), codes are not perfect in general and actually the density of the union of balls of radius $\lfloor \frac{d-1}{2} \rfloor$ is rather small. Roughly speaking there remains many room out of this union of balls. This observation suggests that decoding can be improved and that correcting more than $\lfloor \frac{d-1}{2} \rfloor$ errors might be possible. In short, there is room for improvement.

Example 8.5. If you consider a Reed–Solomon code over $\mathbb{F}_q$ with rate $R = \frac{1}{2}$. Using Berlekamp Welch, one can correct up to $\approx \frac{1-R}{2}n = \frac{n}{4}$ errors. On the other hand, Shannon Theorem asserts the existence of an algorithm correcting almost any error pattern of weight δn where $H_q(\delta) = 1 - R - \varepsilon = \frac{1}{2} - \varepsilon$. For q large enough (which is necessary to have n large when dealing with Reed–Solomon codes), $H_q(\delta) \approx \delta$ and hence Shannon asserts that $\approx \frac{n}{2}$ errors can be corrected, which is twice what Berlekamp welch can perform!

To improve the decoding radius and hence to correct more errors, a solution consists in designing algorithms which are able to return a list of solutions instead of a unique solution. See § 2.1.1 for further details.

For this sake, the radius r should be chosen so that the list is not too large. Indeed, if for instance $r = n$ then the algorithm would return the list of any codeword of the code, which would be stupid. For this reason the decoding radius should be so that the size of the output list is less than polynomial in the code length. This is the point of Johnson bound introduced in §8.2.2.

Before introducing this bound let us first discuss informally the rationale behind list decoding.

8.2.1 Are list-decoding algorithms relevant?

A natural question is: *if I want to correct errors, what could I do with a list of solutions?* There are several answers which lead to a common opinion: *list decoding is relevant.*

1. In the list you receive, you can sort the elements by decreasing distance to the emitted word, and choose the closest which is the solution of the maximum likelihood decoding problem. But what if non unique?

2. If not unique, it depends on the device/situation in which you use codes, may be you can ask the sender to resend the same block.

3. But actually **we don't care** since, in practice, the algorithm turns out to return a list of size $\leqslant 1$ almost all the time.

8.2.2 The Johnson bound

For a list decoding algorithm to be relevant, the size of the list should be bounded above and it is reasonable to expect an upper bound which is polynomial in the code length. The Johnson bound is a bound is a bound on the number of errors that, if satisfies, asserts that the returned list will have at most a polynomial size.

The proof of the following result is omitted and can be found in [Rud].

Theorem 8.6 (Johnson bound)**.** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ be a code of minimum distance $d = \delta n$. Set*

$$\rho \stackrel{def}{=} \left(1 - \frac{1}{q}\right) \cdot \left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right).$$

Then, for any $\boldsymbol{y} \in \mathbb{F}_q^n$,

$$|\{\boldsymbol{c} \in \mathscr{C} \mid d_H(\boldsymbol{y}, \boldsymbol{c}) \leqslant \rho n\}| \leqslant qdn = O(qn^2).$$

The quantity ρ in the previous Theorem is a *relative decoding radius* and is usually referred to as *the Johnson radius*.

Remark 31. For $q = 2$, we get a Johnson radius:

$$\rho_2 = \frac{1}{2}\left(1 - \sqrt{1 - 2\delta}\right).$$

Remark 32. When q tends to infinity, we get an asymptotic Johnson radius:

$$\rho_\infty = 1 - \sqrt{1 - \delta}.$$

In particular for a Reed–Solomon of large length, the base field should be large, and hence q should be large. And since such a code is MDS, we get a radius

$$\rho = 1 - \sqrt{R}.$$

From now on, this asymptotic relative decoding radius is our target

8.2.3 Sudan algorithm

In 1997 Madhu Sudan proposed a polynomial time list decoding algorith for Reed–Solomon codes. This result was an impressive breakthrough in the area of algebraic coding theory for which Sudan received the Nevanlinna award.

Notation 8.1. For a two variables polynomial $P \in \mathbb{F}_q[X, Y]$, the total degree of P is referred to as $\deg P$ while its degree as a polynomial in X (resp. Y) is referred to as $\deg_X P$ (resp. $\deg_Y P$). For instance the polynomial $P := 1 + 2X + 3X^2Y$ satisfies

$$\deg P = 3, \quad \deg_X P = 2, \quad \text{and} \quad \deg_Y P = 1.$$

One can reformulate Berlekamp Welch algorithm as follows. Set $t \stackrel{\text{def}}{=} \frac{n-k}{2}$ which is supposed to be an upper bound on the number of errors.

Step 1. Interpolation Construct a polynomial $Q \in \mathbb{F}_q[X, Y]$ of degree 1 in y of the form $Q = Q_0(X) + Q_1(X)Y$ with $\deg Q_0 < n - t$ and $\deg Q_1 \leqslant n - k - t$ such that

$$\forall i \in \{1, \ldots, n\}, \quad Q(x_i, y_i) = 0.$$

Step 2. Root finding Compute the root of Q as a polynomial in y, that is compute $-\frac{Q_0}{Q_1}$.

The algorithm works since if $f \in \mathbb{F}_q[X]_{<k}$ is such that $(f(x_1), \ldots, f(x_n))$ is the transmitted word and hence has distance less than t to $\boldsymbol{y}$, then the univariate polynomial $Q(X, f(X))$ is zero. Indeed, $\deg Q_0(X) + Q_1(X)f(X) < n - t$ while this polynomial vanishes a at least $n - t$ of the x_i's since for all the indexes i where no error occured, $y_i = f(x_i)$ and hence $Q(x_i, f(x_i)) = Q(x_i, y_i) = 0$.

The key idea of Sudan is that, to correct more errors than $\frac{d-1}{2}$, we need to allow the algorithm to return more than one unique solution. Since the solutions are computed as roots (with respect to the variable Y) of $Q(X, Y)$, the Y–degree of Q should be larger than 1. Here is the core of Sudan's algorithm. Assume we received a word $\boldsymbol{y}$ and we seek all the words $\boldsymbol{c} \in \mathbf{RS}_k(\boldsymbol{x})$ at distance less than t from $\boldsymbol{y}$. The optimal integer t will be determined further.

Step 1. Interpolation. First compute a polynomial

$$Q = \sum_{i=0}^{\ell} Q_i(X) Y^i$$

such that

(S1) $\forall i$, $\deg Q_i + i(k-1) < n - t$;

(S2) $\forall j \in \{1, \ldots, n\}$, $\quad Q(x_j, y_j) = 0$.

Step 2. Root finding. Compute all the polynomials $f \in \mathbb{F}_q[X]_{<k}$ such that $Q(X, f(X)) \equiv 0$.

The first step consists in solving a system of linear equations. The variables of the system are the coefficients of Q and the equations are given by the interpolation conditions. For the second step, there is **no necessity** to perform a complete factorization of Q since, only the factors of Y–degree 1 worth. These factors can be computed by Newton or Newton–Puiseux method.

Remark 33. In practice, the most costly part of the algorithm is the first part: solving a linear system.

The following lemma asserts that every solution of our problem is returned by the algorithm.

Lemma 8.7. *Let Q be a bivariate polynomial satisfying conditions (S1) and (S2). Let $f \in \mathbb{F}_q[X]_{<k}$ such that $d_H(\boldsymbol{y}, (f(x_1), \ldots, f(x_n))) \leqslant t$. Then*

$$Q(X, f(X)) \equiv 0.$$

Proof. Condition (1) on degrees asserts that

$$Q(X, f(X)) = \sum_i Q_i f^i(X)$$

has degree $< n - t$. Moreover, since $\mathrm{d_H}(\boldsymbol{y}, (f(x_1), \ldots, f(x_n))) \leqslant t$, then for at least $n - t$ indexes j, we have $f(x_j) = y_j$. Hence, because of condition (2), for at least $n - t$ indexes j, we have $Q(x_j, f(x_j)) = 0$. Therefore, the univariate polynomial $Q(X, f(X))$ has degree $< n - t$ and at least $n - t$ roots. Consequently, this univariate polynomial is zero. □

Remark 34. Notice that the algorithm may return polynomials which do not satisfy the condition $\mathrm{d_H}(\boldsymbol{y}, (f(x_1), \ldots, f(x_n))) \leqslant t$. On the other hand, the point is that all the polynomials which satisfy this inequality are returned.

Thus, Lemma 8.9 asserts that any solution of the list decoding problem is returned by the algorithm. On the other hand, the size of the list is bounded above by the degree in Y of the polynomial Q.

Decoding radius of Sudan algorithm

For the algorithm to work, the polynomial Q computed in the first step should be nonzero. Since this polynomial is obtained as from the resolution of a linear system. If the system has a number of variables which exceeds that of equations, then there is a nontrivial solution. The system has n equations given by the n interpolating conditions (2).

On the other hand, the number of variables are given by the degree conditions (1). Thus the number if variables is

$$v \stackrel{\text{def}}{=} \sum_{i=0}^{\deg_Y(Q)} n - t - i(k-1).$$

Therefore, the maximum possible Y–degree of Q is the maximum integer i such that

$$n - t - i(k-1) > 0$$

That is

$$\deg_Y(Q) < \left\lfloor \frac{n-t}{k-1} \right\rfloor.$$

Set $\ell \stackrel{\text{def}}{=} \left\lfloor \frac{n-t}{k-1} \right\rfloor$. Therefore,

$$\begin{aligned} v &= \sum_{i=0}^{\ell} n - t - i(k-1) \\ &= (n-t+1)\ell - (k-1)\frac{\ell(\ell+1)}{2} \end{aligned}$$

Finally, to make sure the linear system computed in Step 1 has a nontrivial solution, we need to have more variables than equations, i.e. n should be less than v:

$$n < (n-t+1)\ell - (k-1)\frac{\ell(\ell+1)}{2} \tag{8.4}$$

Exact computations permit to deduce an exact decoding radius but the complete calculation is cumbersome. To get some intuition of this decoding radius, let us finish with an asymptotic analysis of this radius.

Caution. Compared to the asymptotic analyses done in Chapter 4 which were done for a fixed base field, here, since we want the length of the codes to tend to infinity and that the length of a Reed Solomon code is bounded above by q, then we need to consider various base fields. Therefore, in the following analysis, the size q of the base field tends to infinity.

Consider a sequence of Reed Solomon codes whose length tends to infinity and with constant rate R. Let $\rho \overset{\text{def}}{=} \frac{r}{n}$ be the relative decoding radius. Then the asymptotic list size ℓ is equivalent to

$$\ell \sim \frac{1-\rho}{R}$$

Next, (8.4) gives (after dividing by n and for $n \to +\infty$)

$$\begin{aligned} 1 &\lessapprox (1-\rho)\frac{1-\rho}{R} - R \cdot \frac{1}{2} \cdot \left(\frac{1-\rho}{R}\right)^2 \\ 1 &\lessapprox \frac{(1-\rho)^2}{2R} \\ \rho &\lessapprox 1 - \sqrt{2R}. \end{aligned}$$

The comparison between Sudan radius and Berlekamp Welch radius is illustrated by Figure 8.1

One observes in particular that this algorithm represents an improvement only for low rates. More precisely The rate should be less than ≈ 0.17 for Sudan algorithm to correct more errors than Berlekamp Welch. On the other hand for rates close to zero, it corrects almost twice more errors.

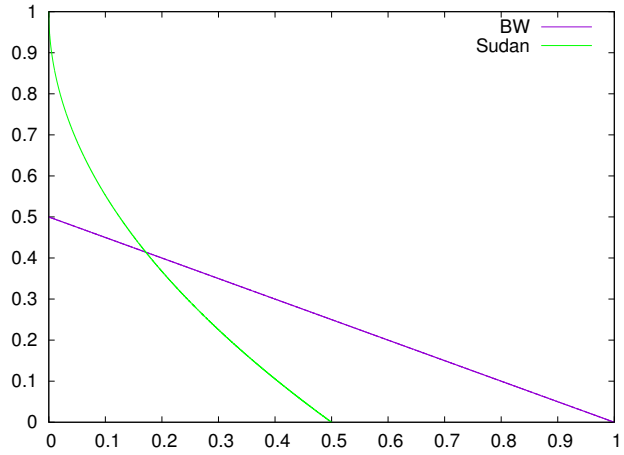


Figure 8.1: Comparison between Sudan and Berlekamp Welch decoding radii

Complexity of Sudan algorithm

According to Remark 33, assuming that the costly part of the algorithm turns out to be the linear algebraic part, then, this part reduces to solve a linear system of n equations and

about n unknowns, which leads to a complexity of $O(n^3)$ operations in $\mathbb{F}_q$.

8.2.4 Decoding up to the Johnson radius : Guruswami Sudan algorithm

In [GS99], Guruswami and Sudan proposed a generalised version of the algorithm which turns out to correct errors up to the Johnson bound. The major input in the algorithm is to enhance the interpolating part by adding some *multiplicity constraints*.

Definition 8.8. Let $Q \in \mathbb{F}_q[X,Y]$ be a polynomial. One says that Q vanishes at $(0,0)$ with multiplicity m if the smallest degree of a monomial of Q is m. One says that Q vanishes at $(a,b) \in \mathbb{F}_q^2$ with multiplicity m if $Q(X-a,Y-b)$ vanishes at $(0,0)$ with multiplicity m. Equivalently, it means that the Taylor expansion of Q in the variables $X-a, Y-b$ is of the form:

$$Q(X,Y) = \lambda_0(X-a)^m + \lambda_1(X-a)^{m-1}(Y-b) \\ + \cdots + \lambda_{m-1}(X-a)(Y-b)^{m-1} + \lambda_m(Y-b)^m + R(X-a,Y-b),$$

where R has only monomials of degree $> m$ and at least one of the λ_i's is nonzero.

Remark 35. Another description of a polynomial $Q \in \mathbb{F}_q[X,Y]$ vanishing at $(0,0)$ (resp. (a,b)) with multiplicity m is that any of its monomials $a_{i,j}X^iY^j$ (resp. $b_{i,j}(X-a)^i(Y-b)^j$) have total degree $i+j \geqslant m$. In particular if the polynomial vanishes at (a,b) with multiplicity m then it can be written as

$$Q(X,Y) = \sum_{j=0}^{m} (X-a)^j (Y-b)^{m-j} Q_j(X,Y) \tag{8.5}$$

for some polynomials $Q_0, Q_1, \ldots, Q_m \in \mathbb{F}_q[X,Y]$.

As said above, Guruswami Sudan algorithm is almost the same as Sudan algorithm with the additional constraint that the interpolating polynomial Q should vanish at the (x_i, y_i) with mutliplicity m for some positive integer m.

The algorithm runs as follows (as in Sudan's algorithm, the algorithm depends on the decoding radius t which will be determined further):

Step 1. Interpolation. First compute a polynomial

$$Q = \sum_{i=0}^{\ell} Q_i(X) Y^i$$

such that

(GS1) $\forall i$, $\deg Q_i + i(k-1) < m(n-t)$;

(GS2) $\forall j \in \{1, \ldots, n\}$, Q vanishes at (x_j, y_j) with multiplicity m.

Step 2. Root finding. Compute all the polynomials $f \in \mathbb{F}_q[X]_{<k}$ such that $Q(X, f(X)) \equiv 0$.

The following lemma is the counterpart of Lemma 8.9.

Lemma 8.9. *Let Q be a bivariate polynomial satisfying conditions (GS1) and (GS2). Let $f \in \mathbb{F}_q[X]_{<k}$ such that $d_H(\boldsymbol{y}, (f(x_1), \ldots, f(x_n))) \leqslant t$. Then*

$$Q(X, f(X)) \equiv 0.$$

Proof. Thanks to condition (GS1), one sees easily that the univariate polynomial $Q(X, f(X))$ has degree less than $m(n-t)$. Moreover, from condition (GS2) and since

$$\mathrm{d_H}(\boldsymbol{y}, (f(x_1), \ldots, f(x_n))) \leqslant t,$$

the polynomial $Q(X, f(X))$ vanishes at least at $n - t$ of the x_i's. Let us prove that if $f(x_i) = y_i$, then $Q(X, f(X))$ vanishes at x_i with multiplicity at least m. From Condition (2) and thanks to (8.5), Q can be written as

$$Q(X, Y) = \sum_{j=0}^{m} (X - x_i)^j (Y - y_i)^{m-j} Q_j(X, Y)$$

for some polynomials $Q_0, \ldots, Q_m$. Moreover, since $f(x_i) = y_i$, the polynomial $f(X) - y_i$ vanishes at x_i and hence

$$f(X) - y_i = (X - x_i) g(X)$$

for some $g \in \mathbb{F}_q[X]$. Therefore,

$$\begin{aligned} Q(X, f(X)) &= \sum_{j=0}^{m} (X - x_i)^j (X - x_i)^{m-j} g(X)^{m-j} Q_j(X, f(X)) \\ &= (X - x_i)^m \sum_{j=0}^{m} g(X)^{m-j} Q_j(X, f(X)). \end{aligned}$$

Therefore, $(X - x_j)^m$ divides $Q(X, f(X))$ and hence this polynomial vanishes at x_i with multiplicity at least m.

In summary, the polynomial $Q(X, f(X))$ has degree $< m(n-t)$ and has at least $n - t$ roots, each one with multiplicity $\geqslant m$. Hence it is identically zero. □

Decoding radius of Guruswami Sudan algorithm

A major change in the radius analysis is the number of equations. Requiring the vanishing of a polynomial $Q \in \mathbb{F}_q[X, Y]$ at a couple $(a, b) \in \mathbb{F}_q^2$ imposes one linear equation that the coefficients of Q. Next requiring its vanishing at (a, b) with multiplicity m imposes $\frac{m(m+1)}{2}$ equations on the coefficients of Q. Indeed, expanding Q as

$$Q = \sum_{i,j} q_{i,j} (X - a)^i (Y - b)^j$$

then any coefficient $q_{i,j}$ with $i+j<m$ should vanish. Hence this imposes $1+2+\cdots+m=\frac{m(m+1)}{2}$ linear conditions on the coefficients of Q.

Consequently, the number of linear equations is:

$$n\frac{m(m+1)}{2}.$$

The number of variables is

$$v \stackrel{\text{def}}{=} \sum_{i=0}^{\deg_Y(Q)} m(n-t)-i(k-1).$$

Therefore, the maximum possible Y–degree of Q is the largest integer i such that

$$m(n-t)-i(k-1)\geqslant 0.$$

Hence

$$\deg_Y Q \leqslant \ell \stackrel{\text{def}}{=} \left\lfloor \frac{m(n-t)}{k-1} \right\rfloor.$$

An approximative analysis in the similar spirit as in Sudan's case gives:

$$\sum_{i=0}^{\lfloor \frac{m(n-t)}{k-1} \rfloor} m(n-t)-i(k-1) > n\frac{m(m+1)}{2} \tag{8.6}$$

$$\frac{m^2(n-t)^2}{k-1}-\frac{1}{2}\frac{m^2(n-t)^2}{(k-1)^2}\cdot(k-1) \gtrapprox n\frac{m(m+1)}{2} \tag{8.7}$$

$$\frac{m^2(n-t)^2}{2(k-1)} \gtrapprox n\frac{m(m+1)}{2} \tag{8.8}$$

$$(n-t)^2 \gtrapprox n\frac{m+1}{m}(k-1) \tag{8.9}$$

Dividing both sides by $\frac{n^2m^2}{2}$ and setting $\rho \stackrel{\text{def}}{=} \frac{t}{n}$, we get:

$$\frac{(1-\rho)^2}{R} \gtrapprox \frac{m+1}{m}$$

$$\rho \lessapprox 1-\sqrt{\frac{m+1}{m}R}.$$

Consequently, when m tends to infinity, we get a radius of the form

$$\rho \lessapprox 1-\sqrt{R}$$

which yields Johnson bound.

To summarize, a comparison between the decoding radii of Berlekamp Welch, Sudan and Guruswami Sudan algorithm is proposed in Figure 8.2

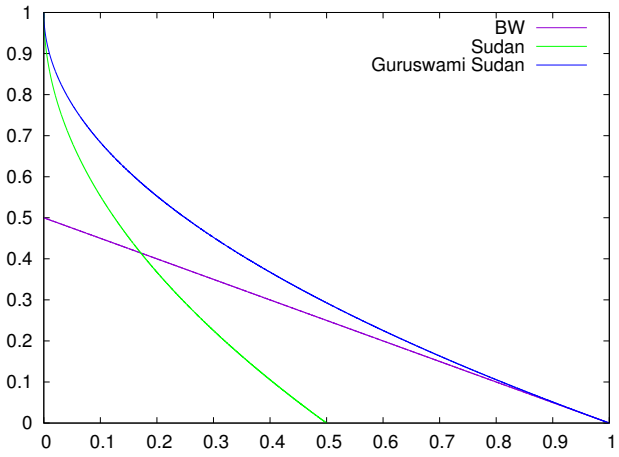


Figure 8.2: Comparison between Berlekamp Welch, Sudan and Guruswami Sudan radii

Complexity of Guruswami Sudan

Here again, let us only consider the cost of the linear algebra. For Guruswami Sudan, the size of the linear system is much larger, since we have $O(nm^2)$ equations and unknowns. Therefore the complexity is $O(n^3m^6)$. Moreover, according to (8.9), for $n(k-1)\frac{m+1}{m}$ to be $\approx n(k-1)$, we need to have $m \approx nk = O(n^2)$ which yields a total complexity of $O(n^{15})$!

Clearly, this algorithm is of theoretical interest, since it asserts that correcting up to the Johnson bound in polynomial time is possible. However, the exponent makes practical implementations hopeless. On the other hand, it is also possible to reduce the multiplicity and get a much smaller complexity at the cost of a shorter decoding radius.

Chapter 9

Cyclic codes and BCH codes

The family of cyclic codes is another fascinating facet of coding theory. Cyclic codes have been subject to intense study since they are of deep interest from theoretical and practical point of views. Indeed, for practical point of view:

- they have a very compact representation since a single row of the generator matrix is sufficient to represent the whole code.
- They benefit from very efficient encoding algorithms using fast Fourier transform.

On the other hand, on the theoretical side cyclic codes, have a wonderfully rich algebraic structure which will be described in what follows.

Let us start with a definition.

Definition 9.1. Let $\sigma : \mathbb{F}_q^n \to \mathbb{F}_q^n$ be the *cyclic shift*:

$$\sigma : \left\{ \begin{array}{ccc} \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ (x_0, \ldots, x_{n-1}) & \longmapsto & (x_{n-1}, x_0, \ldots, x_{n-2}) \end{array} \right. .$$

A code $\mathscr{C}$ is said to be cyclic if it is stable by σ that is to say: $\sigma(\mathscr{C}) = \mathscr{C}$.

Caution. In the previous chapters the vectors were indexe from 1 to n as $(x_1, \ldots, x_n)$. In the present chapter it is more convenient (for a reason which will naturally appear further) to index them from 0 to $n-1$ as $x_0, \ldots, x_{n-1}$.

9.1 First examples

For sure, trivial codes such as the zero code, the full code $\mathbb{F}_q^n$, the parity codes and the repetition codes are cyclic. Now, let us give less trivial examples of cyclic codes.

Lemma 9.2. *Let $\alpha \in \mathbb{F}_q$ be a generator of the multiplicative group $\mathbb{F}_q^\times$. Let $\boldsymbol{x} = (1, \alpha, \alpha^2, \ldots, \alpha^{q-2})$. Then, for any $k \leqslant q-1$, the code $\mathbf{RS}_k(\boldsymbol{x})$ is cyclic.*

Proof. Let $f \in \mathbb{F}_q[x]$ such that $\deg(f)$ and $\boldsymbol{c} = (f(1), f(\alpha), \ldots, f(\alpha^{q-2})) \in \mathbf{RS}_k(\boldsymbol{x})$ be the corresponding codeword. Let $g(X) \stackrel{\text{def}}{=} f(\alpha X)$. One can check that

$$\boldsymbol{c}' = (g(1), g(\alpha), \ldots, g(\alpha^{q-1})) = \sigma(\boldsymbol{c}).$$

Since $g \in \mathbb{F}_q[X]$ with $\deg(g) < k$, then $\sigma(\boldsymbol{c}) \in \mathbf{RS}_k(\boldsymbol{x})$ and this holds for any $\boldsymbol{c} \in \mathbf{RS}_k(\boldsymbol{x})$. □

Lemma 9.3. *Let p be a prime number and $\boldsymbol{x} = (0, 1, \ldots, p-1)$. Then, for any $k \leqslant q-1$, the code $\mathbf{RS}_k(\boldsymbol{x})$ is cyclic.*

Proof. Let $f \in \mathbb{F}_q[x]$ such that $\deg(f)$ and $\boldsymbol{c} = (f(0), f(1), \ldots, f(p-1)) \in \mathbf{RS}_k(\boldsymbol{x})$ be the corresponding codeword. The proof is similar to that of Lemma 9.2 using $g(X) \stackrel{\text{def}}{=} f(X+1)$. □

9.2 The algebraic structure of cyclic codes

9.2.1 Polynomial representation

To understand the algebraic structure behind cyclic codes we use the following polynomial representation of codewords: a codeword $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ is canonically associated to the polynomial $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$.

More formally, there is an $\mathbb{F}_q$–vector space isomorphism

$$\left\{ \begin{array}{ccc} \mathbb{F}_q[X]/(X^n-1) & \longrightarrow & \mathbb{F}_q^n \\ c(X) = \sum_{i=0}^{n-1} c_i X^i & \longmapsto & (c_0, \ldots, c_{n-1}) \end{array} \right. . \tag{9.1}$$

Note that the left–hand term is not written as $\mathbb{F}_q[X]_{<n} \stackrel{\text{def}}{=} \{P \in \mathbb{F}_q[X] \mid \deg P < n\}$ but as the quotient ring $\mathbb{F}_q[X]/(X^n-1)$. As vector spaces, $\mathbb{F}_q[X]_{<n}$ and $\mathbb{F}_q[X]/(X^n-1)$ are isomorphic. On the other hand, $\mathbb{F}_q[X]/(X^n-1)$ has a richer structure: it is a ring. This structure of ring is one of the keys of the study of cyclic codes. Indeed, the cyclic shift corresponds in $\mathbb{F}_q[X]/(X^n-1)$ to the multiplication by X. Formally, the following diagram commutes:

$$\begin{array}{ccc} \mathbb{F}_q[X]/(X^n-1) & \xrightarrow{\sim} & \mathbb{F}_q^n \\ \downarrow{\scriptstyle \times X} & & \downarrow{\scriptstyle \sigma} \\ \mathbb{F}_q[X]/(X^n-1) & \xrightarrow{\sim} & \mathbb{F}_q^n \end{array}$$

Thus, cyclic codes, which are subspaces of $\mathbb{F}_q^n$ which are stable by σ correspond under this isomorphism to subspaces of $\mathbb{F}_q[X]/(X^n-1)$ stable by multiplication by X. Note that a subspace of $\mathbb{F}_q[X]/(X^n-1)$ which is stable by multiplication by X is also stable by multiplication by any element of the ring and hence is an **ideal** of $\mathbb{F}_q[X]/(X^n-1)$. This is summarized by the following statement.

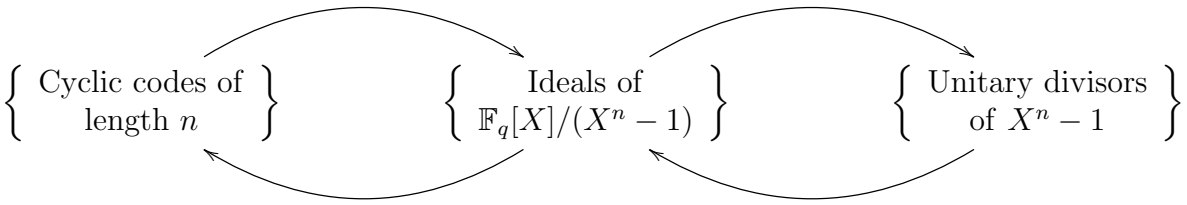
Proposition 9.4. *The isomorphism (9.1) induces a one–to–one correspondence between cyclic codes of $\mathbb{F}_q^n$ and ideals of $\mathbb{F}_q[X]/(X^n-1)$.*

And the previous statement can be turned into a more explicit one.

Proposition 9.5. *The ideals of $\mathbb{F}_q[X]/(X^n-1)$ are in one–to–one correspondence with unitary (with leading coefficient equal to 1) divisors of X^n-1.*

Proof. See appendix B. This is a direct consequence of Propositions B.1 and B.2. □

As a conclusion, we get the following one–to–one correspondences.



Example 9.6. Let us consider the list of cyclic codes of length 7 over $\mathbb{F}_2$. The decomposition of X^7-1 into irreducible factors is

$$X^7+1=(X+1)\cdot(X^3+X+1)\cdot(X^3+X^2+1).$$

To the polynomial $1+X$ corresponds the code with generator matrix

$$\begin{pmatrix}1&1&0&0&0&0&0\\0&1&1&0&0&0&0\\0&0&1&1&0&0&0\\0&0&0&1&1&0&0\\0&0&0&0&1&1&0\\0&0&0&0&0&1&1\end{pmatrix},$$

which is nothing but the parity code of length 7 over $\mathbb{F}_2$.

To the polynomial $1+X+X^3$ corresponds the code with generator matrix:

$$\begin{pmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&0&0&1&1&0&1\end{pmatrix}.$$

And so on...

9.2.2 First consequences for cyclic codes

Example 9.6 suggest two phenomena:

- A generator matrix of a cyclic code associated to a polynomial P can be obtained by stacking the vector representation of P and some of its iterated cyclic shifts.

- The larger the degree of the generating polynomial, the smaller the dimension of the code.

Let us prove these facts.

Theorem 9.7. *Let $\mathscr{C}$ be a cyclic code of length n and dimension k over $\mathbb{F}_q$. There exists a codeword $\boldsymbol{c} \in \mathscr{C}$ called* generator *such that $\boldsymbol{c}, \sigma(\boldsymbol{c}), \sigma^2(\boldsymbol{c}), \ldots, \sigma^{k-1}(\boldsymbol{c})$ is a basis of the code.*

Proof. Consider a generator matrix for C and perform Gaussian elimination in order to construct a nonzero codeword whose last $k-1$ entries are 0. Call

$$\boldsymbol{c} = (c_0 \ c_1 \ \ldots \ c_{n-k} \ 0 \ \ldots \ 0)$$

this codeword. The codewords $\boldsymbol{c}, \sigma(\boldsymbol{c}), \ldots, \sigma^{k-1}(\boldsymbol{c})$ are linearly independent. Indeed, they form an echelonized basis:

$$\begin{pmatrix} c_0 & \cdots & c_{n-k} & 0 & \cdots & 0 \\ 0 & c_0 & \cdots & c_{n-k} & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & c_0 & \cdots & c_{n-k} \end{pmatrix}. \tag{9.2}$$

□

Remark 36. From the codeword $\boldsymbol{c}$ in the above proof, one deduce a polynomial $c(X) = c_0 + c_1 X + \cdots + c_{n-k} X^{n-k}$. This polynomial turns out to be generator of the corresponding ideal of $\mathbb{F}_q[X]/(X^n - 1)$ and hence is a divisor of $X^n - 1$.

Therefore, the above proof, explain how to construct explicitely a generating polynomial of a cyclic code and get a *circulant* generator matrix, i.e. a generator matrix whose rows are obtained by iterating the cyclic shift on the first one. In particular such a matrix is entirely defined by its first row. This has important practical consequences for the storage of such codes: only n bits are necessary to represent any binary cyclic code of length n.

Another consequence of the previous theorem is:

Corollary 9.8. *Let $\mathscr{C}$ be a cyclic code of dimension k has a generating polynomial of degree $n-k$.*

Proof. Using the proof of Theorem 9.7, one sees that the code has a generating polynomial $c(X)$ of degree less than or equal $n-k$. Suppose that its degree is $< n-k$. Then the family $\boldsymbol{c}, \sigma(\boldsymbol{c}), \ldots, \sigma^{k-1}(\boldsymbol{c}), \sigma^k(\boldsymbol{c})$ would be an echelonized family of vectors in the code, which would contradict the hypothesis on the dimension of the code. □

9.2.3 Duality for cyclic codes

Let us start with a lemma on the behaviour of the cyclic shift. with respect to the inner product

Lemma 9.9. *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$, then*

$$\langle \boldsymbol{x}, \sigma(\boldsymbol{y}) \rangle = \langle \sigma^{-1}(\boldsymbol{x}), \boldsymbol{y} \rangle.$$

Proof.

$$\begin{aligned} \langle \boldsymbol{x}, \sigma(\boldsymbol{y}) \rangle &= x_0 y_1 + x_1 y_2 + \cdots + x_{n-2} y_{n-1} + x_{n-1} y_0 \\ &= x_{n-1} y_0 + x_0 y_1 + x_1 y_2 + \cdots + x_{n-2} y_{n-1} \\ &= \langle \sigma^{-1}(\boldsymbol{x}), \boldsymbol{y} \rangle. \end{aligned}$$

□

The main result on duality is the following theorem.

Theorem 9.10. *The dual of a cyclic code is cyclic.*

Proof. Let $\mathscr{C}$ be a cyclic code and $\boldsymbol{c} \in \mathscr{C}$. Let $\boldsymbol{c}' \in \mathscr{C}^\perp$. Thanks to Lemma 9.9,

$$\langle \boldsymbol{c}, \sigma(\boldsymbol{c}') \rangle = \langle \sigma^{-1}(\boldsymbol{c}), \boldsymbol{c}' \rangle.$$

Since $\mathscr{C}$ is cyclic, then $\sigma^{-1}(\boldsymbol{c}) \in \mathscr{C}$ and hence the above product is zero. Therefore, we prove that for any $\boldsymbol{c} \in \mathscr{C}$,

$$\langle \boldsymbol{c}, \sigma(\boldsymbol{c}') \rangle = 0.$$

Thus, by definition of $\mathscr{C}^\perp$, $\sigma(\boldsymbol{c}') \in \mathscr{C}^\perp$. Since this holds for any $\boldsymbol{c}' \in \mathscr{C}^\perp$, we conclude that it is a cyclic code. □

In addition to the previous theorem, a generating polynomial of $\mathscr{C}^\perp$ can easily be deduced from a generating polynomial of $\mathscr{C}$. To explain this, we first need the following definition.

Definition 9.11. Let $f \in \mathbb{F}_q[X]$ of degree d. The *reciprocal polynomial* $\bar{f}$ of f is the polynomial obtained by reversing the order of the coefficients, i.e:

$$\bar{f} \stackrel{\text{def}}{=} X^k f(1/X).$$

Theorem 9.12. *Let g, h be two polynomials of respective degrees $n-k$ and k satisfying $gh = X^n - 1$. Then the cyclic codes associated to g and $\bar{h}$ are dual to each other.*

Proof. Let us denote by $\mathscr{C}(g)$ and $\mathscr{C}(\bar{h})$ these two codes. Note first that, since $gh = X^n - 1$

$$\deg g + \deg \bar{h} = \deg g + \deg h = n.$$

Therefore, according to Corollary 9.8, we get that

$$\dim \mathscr{C}(g) + \dim \mathscr{C}(\bar{h}) = n$$

and hence, we only have to prove the orthogonality between the two codes. Let k be the degree of h. If $k = 0$ then, $h = \bar{h} = 1$ and $g = X^n - 1$ and the codes $\mathscr{C}(g)$ and $\mathscr{C}(\bar{h})$ are

respectively $\{0\}$ and $\mathbb{F}_q^n$ which are dual to each other. Thus, from now on, one can assume that $k > 1$.

Set

$$\begin{aligned} \boldsymbol{c}_g &\stackrel{\text{def}}{=} (g_0\ g_1\ \cdots\ g_{n-k}\ 0\ \cdots\ 0) \\ \boldsymbol{c}_{\bar{h}} &\stackrel{\text{def}}{=} (h_k\ h_{k-1}\ \cdots\ h_0\ 0\ \cdots\ 0). \end{aligned}$$

From Theorem 9.7, the codes $\mathscr{C}(g)$ and $\mathscr{C}(h)$ are respectively generated by $\boldsymbol{c}_g, \sigma(\boldsymbol{c}_g), \ldots, \sigma^{k-1}(\boldsymbol{c}_g)$ and $\boldsymbol{c}_{\bar{h}}, \sigma(\boldsymbol{c}_{\bar{h}}), \ldots, \sigma^{n-k-1}(\boldsymbol{c}_{\bar{h}})$. It suffices to prove that any element of the first basis is orthogonal to an element of the second one. Let $0 \leqslant i < k$ a,d $0 \leqslant j < n-k$ and suppose first that $j \geqslant i$. We have, from Lemma 9.9,

$$\begin{aligned} \langle \sigma^i(\boldsymbol{c}_g), \sigma^j(\boldsymbol{c}_{\bar{h}}) \rangle &= \langle \boldsymbol{c}_g, \sigma^{j-i}(\boldsymbol{c}_{\bar{h}}) \rangle \\ &= g_{j-i}h_k + g_{j-i+1}h_{k-1} + \cdots g_{j-i+k}h_0 \end{aligned}$$

with the convention $g_s = 0$ for any $s > k$. It is easy to check that the above quantity is nothing but the coefficient of degree $j - i + k$ of $gh = X^n - 1$. Since $j < n - k$, $j \geqslant i$ and $k > 1$, then $0 < j - i + k < n$ and hence this coefficient is zero. Therefore:

$$\langle \sigma^i(\boldsymbol{c}_g), \sigma^j(\boldsymbol{c}_{\bar{h}}) \rangle = 0.$$

The case $j < i$ can be treated in a very similar fashion. □

9.3 The use of roots of unity

From now on, we consider cyclic codes whose length n is prime to the characteristic of the ground field. Thus, the polynomial $X^n - 1 \in \mathbb{F}_q[X]$ has simple roots in a suitable extension of $\mathbb{F}_q$. This suitable extension is referred to as the *n–th cyclotomic extension of* $\mathbb{F}_q$ and is the smallest extension of $\mathbb{F}_q$ containing the n–th roots of 1. Let ζ_n be a primitive root of unity in the algebraic closure of $\mathbb{F}_q$, then $\mathbb{F}_q(\zeta_n)$ is the n–th cyclotomic extension of $\mathbb{F}_q$ and the n–th roots of 1 are nothing but $1, \zeta_n, \zeta_n^2, \ldots, \zeta_n^{n-1}$.

9.3.1 Cyclotomic classes

Remind that our interest lies in the decomposition of $X^n - 1$ as a product of irreducible factors in $\mathbb{F}_q[X]$. Indeed, because of the correspondence between cyclic codes and divisors of $X^n - 1$, the irreducible factors of $X^n - 1$ can be regarded as elementary bricks to construct cyclic codes.

Of course, the decomposition of $X - 1$ in $\mathbb{F}_q(\zeta_n)[X]$ is obvious:

$$X^n - 1 = \prod_{i \in \mathbb{Z}/n\mathbb{Z}} (X - \zeta_n^i).$$

Remark 37. The notation "product over $\mathbb{Z}/n\mathbb{Z}$" makes sense since $\zeta_n^n = 1$ and hence, for any integer i, ζ_n^i depends only on the class of i modulo n.

The point is that the factors $(X - \zeta_n^i)$ are not defined over $\mathbb{F}_q$ in general, thus we need to find products of such factors which are in $\mathbb{F}_q[X]$. The following lemma is useful for this purpose.

Lemma 9.13. *Let $\mathbb{F}_{q^\ell}$ be a finite extension of $\mathbb{F}_q$ and $\alpha_1, \ldots, \alpha_m$ be a tuple of distinct elements of $\mathbb{F}_{q^\ell}$. Then the polynomial*

$$P \stackrel{def}{=} \prod_{i=1}^{m} (X - \alpha_i)$$

is in $\mathbb{F}_q[X]$ if and only if for all $i \in \{1, \ldots, m\}$, we have $\alpha_i^q \in \{\alpha_1, \ldots, \alpha_m\}$.

Proof. If $P \in \mathbb{F}_q[X]$, i.e. $P = \sum_{i=0}^{m} p_i X^i$ where the p_i's are in $\mathbb{F}_q$ then, for all i, we have $p_i = p_i^q$ and hence

$$\begin{aligned} P(X^q) &= \sum_{i=0}^{m} p_i (X^q)^i = \sum_{i=0}^{m} p_i^q (X^q)^i \\ &= \sum_{i=0}^{m} (p_i X^i)^q = \left(\sum_{i=0}^{m} p_i X^i \right)^q \\ &= P(X)^q. \end{aligned}$$

Therefore, if $\alpha \in \mathbb{F}_{q^\ell}$ is a root of P, then $P(\alpha) = 0 = P(\alpha)^q = P(\alpha^q)$. Hence, α^q is also a root of P.

Conversely, if the set of roots of P is closed under the Frobenius map, then, since, the coefficients of $P = \sum_{i=0}^{m} p_i X^i$ are the elementary symmetric polynomials:

$$\begin{aligned} p_0 &= (-1)^m \alpha_1 \alpha_2 \cdots \alpha_m \\ &\vdots \\ p_{m-2} &= (-1)^{m-1} \sum_{1 \leqslant i < j \leqslant m} \alpha_i \alpha_j \\ p_{m-1} &= -\alpha_1 - \cdots - \alpha_m \\ p_m &= 1 \end{aligned}$$

Then, one checks that, since any of these coefficients are symmetric polynomials in the $\alpha_i's$, then for all j, $p_j^q = p_j$ and hence, $P \in \mathbb{F}_q[X]$. □

Corollary 9.14. *A factor of $X^n - 1$ in $\mathbb{F}_q[X]$ is of the form*

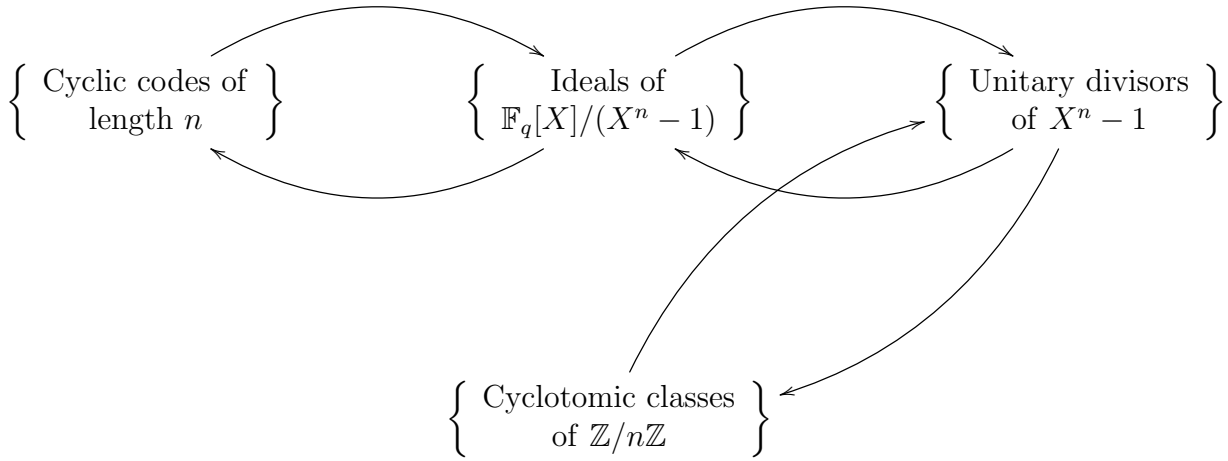
$$g = \prod_{i \in I} (X - \zeta_i)$$

where $I \subset \mathbb{Z}/n\mathbb{Z}$ is stable by multiplication by q.

Definition 9.15. A *cyclotomic class* is a subset of $\mathbb{Z}/n\mathbb{Z}$ which is stable by multiplication by q. A nonempty cyclotomic class is said to be *minimal* if it is minimal for inclusion, that is: if any nonempty subset is not a cyclotomic class.

The point of this definition is that cyclotomic classes are in one-to-one correspondence with factors of $X^n - 1$ in $\mathbb{F}_q[X]$. Moreover, one checks easily that minimal cyclotomic classes are in one-to-one correspondence with irreducible factors of $X^n - 1$ in $\mathbb{F}_q[X]$.

This observation permits to complete the picture page 91:



Example 9.16. For $q = 2$ and $n = 17$, the minimal cyclotomic classes are:

$$\{0\}, \{1, 2, 4, 8, 16, 15, 13, 9\}, \{3, 6, 12, 7, 14, 11, 5, 10\}$$

This entails that $X^{17} - 1$ has 3 irreducible factors over $\mathbb{F}_2$, namely:

$$1 + X, 1 + X^3 + X^4 + X^5 + X^8, \text{ and } 1 + X + X^2 + X^4 + X^6 + X^7 + X^8.$$

Moreover, the complete list of cyclotomic classes is:

$$\begin{aligned}&\emptyset, \{0\}, \{1, 2, 4, 8, 16, 15, 13, 9\}, \{3, 6, 12, 7, 14, 11, 5, 10\},\\ &\{0, 1, 2, 4, 8, 16, 15, 13, 9\}, \{0, 3, 6, 12, 7, 14, 11, 5, 10\}, \text{and } \mathbb{Z}/17\mathbb{Z}.\end{aligned}$$

which correspond respectively to the following divisors of $X^{17} - 1$:

$$\begin{aligned}&1, 1 + X, 1 + X^3 + X^4 + X^5 + X^8, 1 + X + X^2 + X^4 + X^6 + X^7 + X^8,\\ &1 + X + X^3 + X^6 + X^8 + X^9, 1 + X^3 + X^4 + X^5 + X^6 + X^9, \text{and } X^{17} - 1,\end{aligned}$$

which correspond respectively to the following codes.

- Polynomial 1 corresponds to the *full code*: $\mathbb{F}_2^{17}$;

- $X-1$ corresponds to the parity code whose (which is obviously cyclic). It has a generator matrix whose first row is

$$(1\ 1\ 0\ \ \cdots\ 0)$$

and whose other rows are obtained by iterating the cyclic shift on the first one.

- $1+X^3+X^4+X^5+X^8$ corresponds to a cyclic code of dimension 9 having a generator matrix whose first row is:

$$(1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

and the other rows are obtained by iterating the cyclic shift on the first one.

- etc...

- $X^{17}-1$ corresponds to the zero code.

9.3.2 BCH codes

The name "BCH" is due to the fathers of this family of codes: Bose, Chaudhury and Hocquenghem. BCH codes are cyclic codes whose generating polynomial has prescribed roots in geometric progression. That is a sequence of roots of the form

$$\zeta_n^a, \zeta_n^{a+1}, \ldots, \zeta_n^{a+s}.$$

The interest of having roots in geometric progression is that the size of this sequence of roots provides a lower bound for the minimum distance. This is the point of the following result.

Theorem 9.17 (The BCH bound)**.** *Let $\mathscr{C}$ be a cyclic code defined by a cyclotomic class $I \subseteq \mathbb{Z}/n\mathbb{Z}$. Assume that I contains s consecutive integers, $a, a+1, \ldots, a+s-1$, then the minimum distance d of $\mathscr{C}$ satisfies*

$$d \geqslant s+1.$$

Proof. Let $\boldsymbol{c} = (c_0\ \cdots\ c_{n-1}) \in \mathscr{C}$ and $c(X)$ the corresponding polynomial $c(X) = c_0 + c_1X + \cdots + c_{n-1}X^{n-1}$. We have

$$c(\zeta_n^a) = c(\zeta_n^{a+1}) = \cdots = c(\zeta_n^{a+s-1}) = 0. \tag{9.3}$$

Assume that $r \stackrel{\text{def}}{=} \mathrm{w_H}(\boldsymbol{c}) \leqslant s$, then only r of the coefficients of c are nonzero. Let $c_{i_1}, \ldots, c_{i_r}$ be these nonzero coefficients. From (9.3),

$$\begin{array}{ccccccc}
c_{i_0}\zeta_n^{ai_0} & + & \cdots & + & c_{i_{r-1}}\zeta_n^{ai_{r-1}} & = & 0 \\
c_{i_0}\zeta_n^{(a+1)i_0} & + & \cdots & + & c_{i_{r-1}}\zeta_n^{(a+1)i_{r-1}} & = & 0 \\
\vdots & & & & \vdots & & \vdots \\
c_{i_0}\zeta_n^{(a+s-1)i_0} & + & \cdots & + & c_{i_{r-1}}\zeta_n^{(a+s-1)i_{r-1}} & = & 0
\end{array}$$

Which can be re-written as a matrix–vector product:

$$\begin{pmatrix} 1 & \cdots & 1 \\ \zeta_n^{i_0} & \cdots & \zeta_n^{i_{r-1}} \\ \vdots & & \vdots \\ \zeta_n^{(s-1)i_0} & \cdots & \zeta_n^{(s-1)i_{r-1}} \end{pmatrix} \cdot \begin{pmatrix} c_{i_0}\zeta_n^{t_i 0} \\ c_{i_1}\zeta_n^{ti_1} \\ \vdots \\ c_{i_{r-1}}\zeta_n^{ti_{r-1}} \end{pmatrix} = 0.$$

The left-hand matrix is a troncated Van der Monde matrix, i.e. a van der Monde matrix whose last rows have been removed. Since the elements $\zeta_n^{i_0}, \ldots, \zeta_n^{i_{r-1}}$ are distinct, this matrix has full rank and hence his kernel is zero. Thus, $c_{i_0} = c_{i_1} = \cdots = c_{i_{r-1}} = 0$, which is a contradiction. Therefore, no nonzero codeword has weight $\leqslant s$. □

Definition 9.18 (BCH code)**.** Let δ be a positive integer. The BCH code $\mathbf{BCH}_{q,n}(a, \delta)$ is the cyclic code of length n over $\mathbb{F}_q$ associated to the smallest cyclotomic class containing $a+1, a+2, \ldots, a+\delta-1$. If t, the the code is denoted $\mathbf{BCH}_{q,n}(\delta)$. According to Theorem 9.17, such a code has a minimum distance larger than or equal to δ.

Remark 38. Of course, the actual minimum distance of the code $\mathbf{BCH}_{q,n}(a, \delta)$ may exceed δ. The quantity δ is however the *designed* minimum distance and the known algebraic decoding algorithms correct errors up to $\lfloor \frac{\delta-1}{2} \rfloor$.

A first example

Take $q = 2$ and $n = 15$. For $q = 2$, the minimal nonempty cyclotomic classes of $\mathbb{Z}/15\mathbb{Z}$ are:

$$\{0\},\ \{1, 2, 4, 8\},\ \{3, 6, 12, 9\},\ \{5, 10\},\ \{7, 14, 13, 11\}.$$

They correspond respectively to the polynomials

$$g_0 \stackrel{\text{def}}{=} 1 + X,\ g_1 \stackrel{\text{def}}{=} 1 + X + X^4,\ g_3 \stackrel{\text{def}}{=} 1 + X + X^2 + X^3 + X^4, \text{ and } g_5 \stackrel{\text{def}}{=} 1 + X + X^2.$$

Remark 39. The index of each polynomial correspond to the first element of the corresponding cyclotomic class.

Remark 40. Note that the above correspondence may depend on the choice of a primitive n–th root of unity ζ_n.

Then, one can construct the following codes.

Code	Cyclotomic class	Dimension	Correction capability	Polynomial
$\mathbf{BCH}_{2,15}(3)$	$\{1, 2, 4, 8\}$	11	1	g_1
$\mathbf{BCH}_{2,15}(5)$	$\{1, 2, 3, 4, 6, 8, 9, 12\}$	7	2	$g_1 g_3$
$\mathbf{BCH}_{2,15}(7)$	$\{1, 2, 3, 4, 5, 6, 8, 9, 10, 12\}$	5	3	$g_1 g_3 g_5$

Remark 41. The degree of the field extension $[\mathbb{F}_q(\zeta_n) : \mathbb{F}_q]$ is the cardinality of the largest cyclotomic class (left as exercise). Therefore, in the previous example, the cyclotomic extension $\mathbb{F}_2(\zeta_n)$ is nothing but $\mathbb{F}_{16}$.

Reed-Solomon codes of length $q-1$ as particular BCH codes

In the case $n = q-1$, then $\mathbb{F}_q$ contains all $(q-1)$–th roots of unity. Indeed, remind that

$$X^{q-1} - 1 = \prod_{a \in \mathbb{F}_q^\times} (X - a).$$

In this situation, the minimal nonempty cyclotomic classes have a unique element (multiplication by q un $\mathbb{Z}/(q-1)\mathbb{Z}$ is multiplication by 1!). By this manner, one can construct BCH codes with designed distance δ with a cyclotomic class of size $\delta - 1$ and hence have a cyclic code with a generating polynomial of degree $\delta - 1$ and hence a code of length $n = q-1$, dimension $n - \delta + 1$ and minimum distance $\geqslant \delta$. Such a cyclic code is obviously MDS. One can prove that such codes are nothing but Reed-Solomon codes whose support is the set of elements of $\mathbb{F}_q^\times$ ordered as follows:

$$(1, \alpha, \alpha^2, \ldots, \alpha^{q-2})$$

where α is a generator of the multiplicative group $\mathbb{F}_q^\times$.

9.3.3 Decoding BCH codes

See Exercise sheet #4.

Bibliography

[BCG+17] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. *Algorithmes Efficaces en Calcul Formel*. Frédéric Chyzak (auto-édit.), Palaiseau, September 2017. 686 pages. Imprimé par CreateSpace. Aussi disponible en version électronique.

[BDB12] Simeon Ball and Jan De Beule. On sets of vectors of a finite vector space in which every subset of basis size is a basis ii. *Des. Codes Cryptogr.*, 65(1):5–14, 2012.

[BMvT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, 24(3):384–386, May 1978.

[Del75] Philippe Delsarte. On subfield subcodes of modified Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 21(5):575–576, 1975.

[Dem09] Michel Demazure. *Cours d'algèbre*. Cassini, 2nd edition, 2009.

[GS99] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45(6):1757 –1767, sep. 1999.

[Gur10] Venkantesan Guruswami. Introduction to coding theory, Spring 2010. Lecture Notes.
`http://www.cs.cmu.edu/~venkatg/teaching/codingtheory/`.

[HP03] W. Cary Huffman and Vera Pless. *Fundamentals of error-correcting codes*. Cambridge University Press, Cambridge, 2003.

[JH04] Jørn Justesen and Tom Høholdt. *A course in error-correcting codes*. EMS Textbooks in Mathematics. European Mathematical Society (EMS), Zürich, 2004.

[McE78] Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

[Mor91] Carlos Moreno. *Algebraic curves over finite fields*, volume 97 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1991.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes. I.* North-Holland Publishing Co., Amsterdam, 1977. North-Holland Mathematical Library, Vol. 16.

[Rud] Atri Rudra. Error correcting codes: Combinatorics, algorithms and applications. Lecture Notes. `http://www.cse.buffalo.edu/faculty/atri/courses/coding-theory/fall07.html`.

[Sud97] Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180 – 193, 1997.

[TVZ82] M. A. Tsfasman, S. G. Vlăduţ, and Th. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Math. Nachr.*, 109:21–28, 1982.

[VNT07] Serge Vlăduţ, Dmitry Nogin, and Michael Tsfasman. *Algebraic Geometric Codes: Basic Notions*. American Mathematical Society, Boston, MA, USA, 2007.

[Wal00] Judy Walker. *Codes and curves*. American Mathematical Society, 2000. Available on line : `http://www.math.unl.edu/∼jwalker7/papers/rev.pdf`.

[Zém13] Gilles Zémor. Théorie de l'information, 2013. Lecture Notes. `http://www.math.u-bordeaux1.fr/∼gzemor/TI.pdf`.

Appendix A

Complements on probability theory

A.1 Proof of Chernoff bound

To prove Chernoff bound, we need the following lemma.

Lemma A.1. *For all $x > 0$, we have* $\log(1+x) \geqslant \frac{x}{1+\frac{x}{2}}$.

Proof. Indeed, let $f : \mathbb{R}_+ \to \mathbb{R}$ defined as $f(x) = \log(1+x) - \frac{x}{1+\frac{x}{2}}$. Then, $f(0) = 0$ and

$$\forall x \in \mathbb{R}_+,\ f'(x) = \frac{1}{1+x} - \frac{1}{(1+\frac{x}{2})^2}.$$

For all $x \in \mathbb{R}_+$, we have

$$\left(1+\frac{x}{2}\right)^2 = 1 + x + \frac{x^2}{4} \leqslant 1+x \quad \Longleftrightarrow \quad \frac{1}{1+x} - \frac{1}{(1+\frac{x}{2})^2} \leqslant 0.$$

Therefore, f' is negative on $\mathbb{R}_+$, hence f is decreasing on $\mathbb{R}_+$. Since $f(0) = 0$, we conclude that $f(x)$ is negative for all $x \in \mathbb{R}_+$, which concludes the proof. □

Proof of Chernoff Bound. For all positive real number t, we have,

$$\begin{aligned} \mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) &= \mathbb{P}\Big(e^{t\mathrm{w_H}(\boldsymbol{e})} \geqslant e^{t(1+\varepsilon)pn}\Big) \\ &\leqslant \frac{\mathbb{E}\left(e^{t\mathrm{w_H}(\boldsymbol{e})}\right)}{e^{t(1+\varepsilon)pn}}. \end{aligned} \tag{A.1}$$

The last inequality is a direct consequence of Markov inequality. Remind that $\boldsymbol{e} = (e_1, \ldots, e_n)$. For all i, denote by $\mathrm{w_H}(e_i)$ the integer 0 is $e_i = 0$ and 1 if $e_i \neq 0$. We have $\mathrm{w_H}(\boldsymbol{e}) = \sum_{i=1}^n \mathrm{w_H}(e_i)$ and, since the e_i's are independent random variables, so are the random variables $e^{t\mathrm{w_H}(e_i)}$. Hence,

$$\mathbb{E}\left(e^{t\mathrm{w_H}(\boldsymbol{e})}\right) = \mathbb{E}\left(e^{t\sum_{i=1}^n \mathrm{w_H}(e_i)}\right) = \mathbb{E}\left(\prod_{i=1}^n e^{t\mathrm{w_H}(e_i)}\right) = \prod_{i=1}^n \mathbb{E}\left(e^{t\mathrm{w_H}(e_i)}\right), \tag{A.2}$$

where the last equality is a consequence of the independence of the random variables. Moreover, for all $i \in \{1, \dots, n\}$,

$$\mathbb{E}(e^{t\mathrm{w_H}(e_i)}) = (1-p) + pe^t = 1 + p(e^t - 1) \leqslant e^{p(e^t-1)}. \tag{A.3}$$

Therefore, from (A.2) and (A.3),

$$\mathbb{E}\left(e^{t\mathrm{w_H}(\boldsymbol{e})}\right) \leqslant e^{pn(e^t-1)}.$$

Using (A.1), we obtain,

$$\mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant \frac{e^{pn(e^t-1)}}{e^{pnt(1+\varepsilon)}}.$$

Set[1] $t = \log(1+\varepsilon)$, we get,

$$\mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant e^{pn(\varepsilon-(1+\varepsilon)\log(1+\varepsilon))}. \tag{A.4}$$

Combining (A.4) and Lemma A.1, we get

$$\mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant e^{pn\left(-\frac{\varepsilon^2}{2+\varepsilon}\right)}.$$

Since $0 < \varepsilon < 1$, we get the result:

$$\mathbb{P}\Big(\mathrm{w_H}(\boldsymbol{e}) \geqslant (p+\varepsilon)n\Big) \leqslant e^{-\frac{pn\varepsilon^2}{3}}.$$

□

A.2 Entropy and volume of balls

Proof of Lemma 3.11. Let n be a non negative integer. From Newton formula, we have

$$1 = (p + (1-p))^n = \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} \tag{A.5}$$

$$\geqslant \sum_{i=0}^{pn} \binom{n}{i} p^i (1-p)^{n-i}. \tag{A.6}$$

$$\geqslant \sum_{i=0}^{pn} \binom{n}{i} (q-1)^i \left(\frac{p}{q-1}\right)^i (1-p)^{n-i} \tag{A.7}$$

$$\geqslant (1-p)^n \sum_{i=0}^{pn} \binom{n}{i} (q-1)^i \left(\frac{p}{(q-1)(1-p)}\right)^i \tag{A.8}$$

[1] By "log" we mean the Neperian logarithm such that $\log(e) = 1$.

Moreover, since, by assumption, $p \leqslant 1 - \frac{1}{q}$,

$$\frac{p}{(q-1)(1-p)} \leqslant \frac{1-\frac{1}{q}}{(q-1)\frac{1}{q}} = 1,$$

and hence,

$$\forall i \in \{0, \dots, pn\}, \quad \left(\frac{p}{(q-1)(1-p)}\right)^i \geqslant \left(\frac{p}{(q-1)(1-p)}\right)^{pn}. \tag{A.9}$$

Combining (A.8) and (A.9), we get:

$$1 \geqslant (1-p)^n \left(\frac{p}{(q-1)(1-p)}\right)^{pn} \underbrace{\sum_{i=0}^{pn} \binom{n}{i} (q-1)^i}_{= \mathrm{Vol}_q(pn, n)}.$$

and since,

$$(1-p)^n \left(\frac{p}{(q-1)(1-p)}\right)^{pn} = p^{pn}(1-p)^{(1-p)n}(q-1)^{-pn} = q^{-nH_q(p)},$$

we obtain

$$1 \geqslant q^{-nH_q(p)} \mathrm{Vol}_q(pn, n) \quad \text{and hence,} \quad \mathrm{Vol}_q(pn, n) \leqslant q^{nH_q(p)}.$$

This proves (1).

To prove (2), we first use the following obvious fact: "the volume of the ball of radius pn is larger than the volume of the sphere of radius pn." That is:

$$\mathrm{Vol}_q(pn, n) = \sum_{i=0}^{pn} \binom{n}{i} (q-1)^i \geqslant \binom{n}{pn} (q-1)^{pn}. \tag{A.10}$$

Next, using Stirling formula[2],

$$\begin{aligned} \binom{n}{pn} = \frac{n!}{pn!((1-p)n)!} &\sim \frac{\left(\frac{n}{e}\right)^n \sqrt{2n\pi}}{\left(\frac{pn}{e}\right)^{pn} \sqrt{2pn\pi} \cdot \left(\frac{(1-p)n}{e}\right)^{(1-p)n} \sqrt{2(1-p)n\pi}} \\ &\sim \frac{1}{\sqrt{2p(1-p)n\pi}} \cdot p^{-pn}(1-p)^{(1-p)n}. \end{aligned}$$

Therefore,

$$\begin{aligned} \binom{n}{pn}(q-1)^{pn} &\sim \frac{1}{\sqrt{2p(1-p)n\pi}} \cdot p^{-pn}(1-p)^{(1-p)n}(q-1)^{pn} \\ &\sim q^{nH_q(p)} \frac{1}{\sqrt{2p(1-p)n\pi}}. \end{aligned}$$

[2]Recall that Stirling formula asserts that $n! \sim_{n \to +\infty} \left(\frac{n}{e}\right)^n \sqrt{2n\pi}$.

That is to say

$$\binom{n}{pn}(q-1)^{pn} = q^{nH_q(p)}\frac{1}{\sqrt{2p(1-p)n\pi}}(1+\circ(1)) \tag{A.11}$$

$$= q^{n(H_q(p)-\varepsilon)}\left(q^{\varepsilon n}\frac{1}{\sqrt{2p(1-p)n\pi}}(1+\circ(1))\right). \tag{A.12}$$

Moreover, for all $\varepsilon > 0$, we have

$$\lim_{n\longrightarrow+\infty} q^{\varepsilon n}\frac{1}{\sqrt{2p(1-p)n\pi}} = +\infty.$$

and hence, for all $\varepsilon > 0$ there exists a large enough integer n such that,

$$\mathrm{Vol}_q(pn, n) \geqslant q^{n(H_q(p)-\varepsilon)}.$$

□

Remark 42. From (A.10), we also proved that asymptotically, the ball and the sphere have the same volume. This phenomenon of discrete geometry holds in some sense in Euclidean geometry. Indeed, if you consider the unit ball $\mathbb{B}_n$ of $\mathbb{R}^n$:

$$\mathbb{B}_n \stackrel{\text{def}}{=} \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\|_2 \leqslant 1\right\},$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Then, let $0 < \delta < 1$ and

$$S_{\delta,n} \stackrel{\text{def}}{=} \left\{\boldsymbol{x} \in \mathbb{R}^n \mid 1-\delta \leqslant \|\boldsymbol{x}\|_2 \leqslant 1\right\}$$

then, one can prove that for all δ, we have

$$\lim_{n\longrightarrow\infty} \frac{\mathrm{Vol}(\mathbb{B}_n)}{\mathrm{Vol}(S_{\delta,n})} = 1.$$

That is to say, when n tends to infinity, most of the volume of the ball is concentrated "on its surface".

Appendix B

Rings in algebra

Remind that an *abelian group* is a set A with an addition law $+$ which is associatve, commutative, such that A contains a zero element denoted by 0 for this law and any $a \in A$ has an opposite element denoted by $-a$.

A *ring* R is an abelian group with a multiplication law "$\times$" which is associative and distributive with respect to $+$. If the law $\times$ is commutative (this is always what happens in these notes), then the ring is said to be a *commutative ring* and if there is an element denoted by $1 \in R$ such that $\forall a \in R,\ a \times 1 = a$, then the ring is said to be an *unit ring*. Finally, a unit ring in which any nonzero element is invertible with respect to $\times$ is said to be a *field*.

An *ideal* I of a ring R is a sub-group of R with respect to law $+$ which is stable by multiplication by any element of R. Given an ideal I of a ring R, the relation $a \sim b$ if $b - a \in I$ is an equivalence relation and the quotient set R/I is also a ring.

Given an element a of a ring R the set $aR \stackrel{\text{def}}{=} \{a \times r \mid r \in R\}$ is an ideal called the ideal spanned by a. Such an ideal is said to be *principal* and a ring in which every ideal is principal is said to be a *principal ideal ring*. For instance, the following result is well–known.

Proposition B.1. *Let k be a field, then the ring $k[X]$ is a principal ideal ring.*

In chapter 9, we need the following result.

Proposition B.2. *Let R be a principal ideal ring and $I = aR$ for some $a \in R$ be an ideal of R, then R/I is a principal ideal ring. Moreover, any ideal J of R/I is generated by the class $\bar{b} \in R/I$ of an element $b \in R$ such that b divides a. That is to say: ideals of R/I are in one to one correspondence with divisors of b (up to multiplication by an invertible element of R).*

The proof of this proposition reposes on the following result.

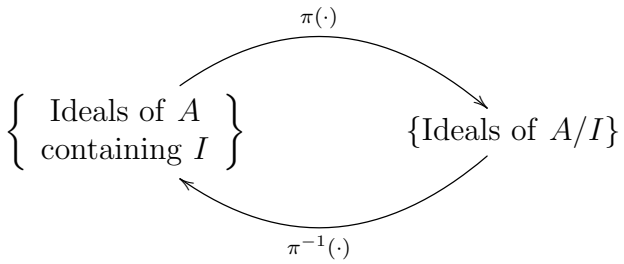
Theorem B.3 (Correspondence of ideals)**.** *Let A be a ring and I an ideal of A. Then, the canonical map $\pi : A \to A/I$ induces a one–to–one correspondence between ideals of A/I and ideals of A containing I. This one–to–one correspondence is explicit: an ideal J of A containing I corresponds to $\pi(J)$, on the other hand, an ideal H of A/I corresponds to $\pi^{-1}(H)$.*

Proof. Remind that the inverse image of an ideal by a morphism of rings is always an ideal. Therefore, for any ideal $H \subseteq A/I$, the inverse image $\pi^{-1}(H)$ is an ideal. On the other hand, since π is a surjective morphism, direct images of ideals by π are ideals.

Second, using again that the map π is surjective, for any ideal H of A/I, we have $\pi(\pi^{-1}(H)) = H$.

Conversely, let J be an ideal of A containing I, then clearly $J \subseteq \pi^{-1}(\pi(J))$. Conversely, let $x \in \pi^{-1}(\pi(J))$, then, by definition of the inverse image, $\pi(x) \in \pi(J)$, thus, $\pi(x) = \pi(a)$ for some $a \in J$. Therefore, $\pi(x - a) = 0$ and hence $x - a \in \ker \pi = I$. Since $a \in J$ and $I \subseteq J$, we conclude that $x \in J$ and hence that $J = \pi^{-1}(\pi(J))$.

Consequently, we have two reciprocal maps:



which proves the one–to–one correspondence. □

Proof of Proposition B.2. By Theorem B.3, the ideals of R/I are in one–to–one correspondence with ideals of R containing I. Since I is the principal ideal spanned by a, the ideals of R/I are in correspondence with ideals of R containing aR. Since R is a principal ideal ring, such ideals are of the form bR with $bR \subseteq aR$ which means that $a = br$ for some $r \in R$ and hence that $b|a$. Finally, for any $b \in R$ such that $b|a$, the ideal $\bar{b}R/I$ is an ideal of R/I and any ideal of R/I can be obtained by this manner. Thus, R/I is a principal ideal ring. □