Department of Mathematics

Ph.D. Programme in Mathematics

XXIX cycle

Optimal Codes and Entropy Extractors

Alessio Meneghetti

UNIVERSITÀ DEGLI STUDI DI TRENTO

2017

University of Trento
Ph.D. Programme in Mathematics

Doctoral thesis in Mathematics

OPTIMAL CODES AND ENTROPY EXTRACTORS

Alessio Meneghetti

Advisor: Prof. Massimiliano Sala

Head of the School: Prof. Valter Moretti

Academic years: 2013–2016

XXIX cycle – January 2017

Acknowledgements

I thank Dr. Eleonora Guerrini, whose Ph.D. Thesis inspired me and whose help was crucial for my research on Coding Theory.

I also thank Dr. Alessandro Tomasi, without whose cooperation this work on Entropy Extraction would not have come in the present form.

Finally, I sincerely thank Prof. Massimiliano Sala for his guidance and useful advice over the past few years.

This research was funded by the Autonomous Province of Trento, Call "Grandi Progetti 2012", project *On silicon quantum optics for quantum computing and secure communications - SiQuro*.

Contents

List of Figures

List of Tables

Introduction

> *Information is the resolution of uncertainty.*

Claude Shannon

> *Truth emerges more readily from error than from confusion.*

Francis Bacon

Coding Theory deals with the problem of safe communication: signals sent through a noisy channel can indeed be corrupted and therefore reach their destination with some errors. In order to protect the information content, some redundancy is added to the message, so that the original data can be retrieved from corrupted signals. The first example of a code is what is commonly known as *repetition code*. Each symbol is repeated several times during transmission, in such a way that the receiver can reconstruct the original message by selecting the most probable one. A fundamental aspect, that arises from the example above, is therefore how to achieve the desired correction capability while keeping the number of redundancy symbols small. This problem is far from simple. We still do not fully know optimal codes, and the study of what is possible to achieve is normally carried on by presenting bounds on code parameters.
Another aspect of communication is privacy. Proofs of security usually rely on the utilisation of randomly picked keys which are used to encrypt the data and have to be kept secret. The first step to allow secure communication is therefore to produce these random keys. The most basic example of *random number generator* is the flip of a balanced coin. In the ideal case, each of the values associated to the two sides of the coin can be generated with equal probability. However, real world generators are not ideal and their outcomes are not balanced. To improve the quality of the random keys we rely on *entropy extractors*, compression functions designed to output random numbers from unbalanced sequences of bits.

In this work we deal with both safety and security of communication as introduced above.
Regarding Coding Theory, we start from a thorough analysis of known bounds on code parameters and a study of the properties of Hadamard codes. We find of particular interest the Griesmer bound, which is a strong result known to be true only for linear codes. We try to extend

it to all codes, and we can determine many parameters for which the Griesmer bound is true also for nonlinear codes. In case of systematic codes, a class of codes including linear codes, we can derive stronger results on the relationship between the Griesmer bound and optimal codes. For example, we prove that the Griesmer bound holds for all binary systematic codes whose distance is either a power of 2 or the difference between two powers of 2. On the other hand, we construct a family of optimal binary systematic codes contradicting the Griesmer bound. Finally, we obtain new bounds on the size of optimal codes.
As for the study of random number generation, we analyse linear extractors and their connection with linear codes. The main result on this topic is a link between code parameters and the entropy rate obtained by a processed random number generator. More precisely, to any linear extractor we can associate the generator matrix of a linear code. Then, we link the total variation distance between the uniform distribution and the probability mass function of a random number generator with the weight distribution of the linear code associated to the linear extractor.
Finally, we present a collection of results derived while pursuing a way to classify optimal codes, such as a probabilistic algorithm to compute the weight distribution of linear codes and a new bound on the size of codes.

This work is organised as follows. Chapters 1 to 4 contain the background on Coding Theory and Random Number Generation needed for the research presented in Chapters 5, 6 and 7.
Chapter 1 describes basic results and definitions on Coding Theory, with focus on nonlinear codes. In Chapter 2 we present an overview of the most known bounds on code parameters, such as the Sphere packing bound, the Gilbert bound, the Varshamov bound, the Singleton bound, the Griesmer bound, the Plotkin bound, the Johnson bound, the Elias bound, the Zinoviev-Litsyn-Laihonen bound and the Bellini-

Guerrini-Sala bounds. Then in Chapter 3 we discuss Hadamard matrices and their link with optimal nonlinear codes. In Chapter 4 we introduce the theory behind Random Number Generation and entropy extraction, following the definitions proposed by NIST.
Chapters 5 to 7 contain results on optimal codes and on entropy extraction. Chapter 5 deals with optimal nonlinear systematic codes. In this chapter we study properties on codes useful to understand the applicability of the Griesmer bound. These properties lead to a classification of parameters for which we can apply the bound. Some examples are:

- all codes whose distance is $d \leq 2q$,
- all codes for which the dimension k and the distance d satisfy $q^{k-1}|d$,
- all binary codes for which the minimum distance is either 2^r or $2^r - 2^s$ for any choice of positive integers $s < r$.

In the same chapter we analyse optimal binary codes with small dimension. Our main results are a complete classification of optimal codes of dimension 2 and a collection of properties of codes of dimension 3. The analysis of optimal codes with respect to the Griesmer bound is then concluded by presenting a new family of optimal nonlinear systematic binary codes contradicting the bound. This family is obtained by using similar methods to the Levenshtein's construction of optimal codes from Hadamard matrices. For each $k > 3$ these codes are $\left(2^{k+1}+2, 2^k, 2^k+2\right)_2$ systematic codes, while the Griesmer bound for codes of dimension k and distance $2^k + 2$ gives length $n \geq 2^{k+1} + k - 1$.
Chapter 6 deals with entropy extractors. Previously known results showed a link between linear extractors and the minimum distance of the codes generated by the same matrix used for the extraction (see e.g. [28] and [29]). We show the connection between the Walsh

spectrum of the output of a binary random number generator and the bias of individual bits, and use this to explain how previously known bounds on the performance of linear binary codes as RNG post-processing functions can be derived as a special case. We then extend this framework to the case of output in non-binary finite fields by use of the Fourier transform.

In Chapter 7 we describe a probabilistic algorithm for computing the weight distribution of linear codes obtained by using methods derived from our study on entropy extractors. Then we present some results on polynomials in $\mathbb{F}_2[X]$, such as a closed formula to compute the binary Möbius transform of a Boolean function and an equivalent formulation of the Hilbert's Nullstellensatz for the particular case of ideals containing the field equations.

Part I

Codes and Random Number Generation

Chapter 1

Introduction to Coding Theory

> *Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages.*
>
> Claude Shannon

In this chapter we introduce all concepts and notations we need to present the main results of this work. Most of the definitions and statements presented here are well known in literature and form the basics of Coding Theory.
The theory presented in this work is not a complete introduction to Coding Theory and we take for granted that readers have a basic knowledge of linear algebra and finite fields. Therefore we list here only some relevant properties and definitions, for a thorough understanding of the subject see e.g. [32]. Regarding Coding Theory, over this entire work we refer to classical books, as for example [22], [23], [33] and [42].

We use as notation q to denote a prime power p^m, $\mathbb{F}_q$ to denote the finite field of size q and $(\mathbb{F}_q)^n$ for the vector space of dimension n over $\mathbb{F}_q$. Whenever not otherwise specified, all vectors are considered to be row vectors. This implies for example that linear maps between vector spaces can be thought as right-multiplications by matrices.
Given two vectors $u, v \in (\mathbb{F}_q)^n$, we denote with $\mathrm{w}(u)$ the Hamming weight of u and with $\mathrm{d}(u, v)$ the Hamming distance between u and v. We often call them simply *weight* and *distance*. Proper definitions of weight and distance are the following.

Definition 1. Let $u \in (\mathbb{F}_q)^n$. The Hamming weight $\mathrm{w}(u)$ is the number of non-zero coordinates of u.

Definition 2. Let u and v be in $(\mathbb{F}_q)^n$. The Hamming distance $\mathrm{d}(u, v)$ between u and v is the number of coordinates in which they differ.

From Definition 2 it follows that $\mathrm{d}(u, v) = \mathrm{w}(u - v)$. On the other hand, $\mathrm{w}(u)$ can be seen as the distance between u and the zero vector.

Definition 3. Given a vector $u \in (\mathbb{F}_q)^n$ of weight s, the *support* of u is the set of s indices $\mathrm{supp}(u) = \{j_1, \ldots, j_s\}$ for which the corresponding coordinates of u are non-zero.

The support of a vector u is therefore linked to its weight by the formula $\mathrm{w}(u) = |\mathrm{supp}(u)|$.
Consider now two vectors $u \in (\mathbb{F}_q)^{n_1}$ and $v \in (\mathbb{F}_q)^{n_2}$. We define the concatenation of u and v as the vector $(u|v) \in (\mathbb{F}_q)^{n_1+n_2}$ whose first n_1 coordinates are the coordinates of u while the last n_2 are the coordinates of v.

Definition 4. A code C is a set of M vectors in the vector space $(\mathbb{F}_q)^n$, where $\mathbb{F}_q$ is the finite field with q elements. We refer to each of these vectors as a *codeword* $c \in C$, to n or $\mathrm{len}(C)$ as the *length* of C and to M or $|C|$ as its *size*.
We denote with $\mathrm{d}(C)$ or simply with d the minimum distance of C, i.e. the minimum among the Hamming distances between any two distinct codewords in C. A code C with such parameters is denoted by the four parameters $(n, M, d)_q$, or as an $(n, M)_q$ code whenever its distance is unknown or not relevant.

Given a code, we can define its weight distribution and distance distribution as follows.

Definition 5. Let C be an $(n, M, d)_q$ code. The *weight distribution* of C is the sequence of integers $\{A_i\}_{i=0,\dots,n}$, where

$$A_i = |\{c \in C \ : \ \mathrm{w}(c) = i\}| \, .$$

In the same way, the *distance distribution* $\{B_i\}_{i=0,\dots,n}$ of C is defined as

$$B_i = |\{(c_1, c_2) \ : \ c_1, c_2 \in C, \ \mathrm{d}(c_1, c_2) = i\}| \, .$$

We say that two codes C and C' are equivalent if there is a bijection $\varphi : C \to C'$ which preserves the distances between any two codewords:

$$C \sim C' \qquad \Longleftrightarrow \qquad \mathrm{d}\left(c_1, c_2\right) = \mathrm{d}\left(\varphi\left(c_1\right), \varphi\left(c_2\right)\right), \ \forall \ c_1, \ c_2 \in C.$$

In particular, translations by fixed vectors, multiplications by permutation matrices and field permutations are maps which preserve the distance between vectors, as stated in Lemma 6.

Lemma 6. *Let θ be a permutation of $\mathbb{F}_q$ and let $\varphi : (\mathbb{F}_q)^n \to (\mathbb{F}_q)^n$ be*

- *a translation by a constant vector $v \in (\mathbb{F}_q)^n$,*
- *the multiplication by a permutation matrix P or*
- *the application of θ to the first coordinate of $(\mathbb{F}_q)^n$.*

Then $C' = \varphi(C)$ is a code equivalent to C.

Proof. By applying Definition 2 it is straightforward to prove the first two items. For the third point, let observe that for any bijection θ we have $\alpha \neq \beta$ if and only if $\theta(\alpha) \neq \theta(\beta)$. □

By combining functions as in Lemma 6 we obtain the following proposition.

Proposition 7. *Let $\theta_1, \ldots, \theta_n$ be permutations of $\mathbb{F}_q$, P an $n \times n$ permutation matrix and $v \in (\mathbb{F}_q)^n$ a chosen vector. Then $\varphi : (\mathbb{F}_q)^n \to (\mathbb{F}_q)^n$ defined by*

$$\varphi(c) = (\theta_1(c_1), \ldots, \theta_n(c_n)) \cdot P + v,$$

with $c = (c_1, \ldots, c_n) \in (\mathbb{F}_q)^n$, is a map which preserves the Hamming distances between vectors.

Let us denote with k the minimum integer allowing the existence of a map F between $(\mathbb{F}_q)^k$ and $(\mathbb{F}_q)^n$ whose image is C itself, and let us call it the *combinatorial dimension* of C. Sometimes we also use $\dim(C)$ to denote the combinatorial dimension of the code C. Clearly, $\dim(C)$ is equal to $\lceil \log_q M \rceil$.
Since F is a vectorial function, we can write

$$F(X) = (f_1(X), \ldots, f_n(X)).$$

Definition 8. We call F a *generator function* of the code. The maps $f_1, \ldots, f_n$ are called *component functions*, or simply *components* of F.

In the general case, the number of possible generator functions of a code is given by

$$\left\{ q^k \atop M \right\} \cdot M!,$$

where $\left\{ a \atop b \right\}$ is the Stirling number of the second kind

$$\left\{ a \atop b \right\} = \frac{1}{b!} \sum_{j=0}^{b} (-1)^{b-j} \binom{b}{j} j^a.$$

In the particular case of $M = q^k$, the number of functions reduces to $M!$.

Definition 9. C is a *linear* code if C is a vector subspace of $(\mathbb{F}_q)^n$. In this case, $M = q^k$ for a certain positive integer k called the *dimension* of the code. Notice that k coincides with the combinatorial dimension of C.
A linear code is denoted as an $[n, k]_q$ code, and with $[n, k, d]_q$ code whenever its distance is known.

The number of generator functions of a linear code is therefore $M! = q^k!$, and among these there is at least a map F for which all $f_1, \ldots, f_n$ are linear maps. In this case F becomes the multiplication by a $k \times n$ matrix G with coefficients in the field, which is known as the *generator matrix* of C.

Example 1. Let C be the $[3, 2, 2]_2$ linear code

$$C = \{c_0, c_1, c_2, c_3\} = \{(0,0,0)\,,\ (1,1,0)\,,\ (0,1,1)\,,\ (1,0,1)\}\,.$$

A possible generator function is given by

$$F(0,0) = (1,1,0),\ F(1,0) = (0,0,0),\ F(0,1) = (0,1,1),\ F(1,1) = (1,0,1),$$

which is the vectorial Boolean function

$$F(x_1, x_2) = (f_1(x_1, x_2), f_2(x_1, x_2), f_3(x_1, x_2))$$

where $f_1,\ f_2$ and f_3 are therefore defined by

$$\begin{array}{lll} f_1(0,0)=1\,, & f_2(0,0)=1\,, & f_3(0,0)=0\,,\\ f_1(1,0)=0\,, & f_2(1,0)=0\,, & f_3(1,0)=0\,,\\ f_1(0,1)=0\,, & f_2(0,1)=1\,, & f_3(0,1)=1\,,\\ f_1(1,1)=1\,, & f_2(1,1)=0\,, & f_3(1,1)=1\,, \end{array}$$

and can be explicitly obtained through the binary Moebius transform, as shown in Section 7.2. We have

$$F(x_1,x_2)=(\ 1+x_1+x_2,\ 1+x_1,\ x_2\ )\,.$$

Notice now that even though C is linear, F is a not a linear function. However, given σ a permutation of $(\mathbb{F}_2)^2$, the image of the function $\bar{F}=F\circ\sigma$ is again a generator function for the same code C, since the image of $\bar{F}$ coincides with the image of F.
By choosing for example $\sigma(x_1,x_2)=(x_1+1,x_2)$ we obtain

$$\bar{F}(x_1,x_2)=(\ x_1+x_2,\ x_1,\ x_2\ )\,,$$

to which is associated the generator matrix

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

Notice that in example 1 the map F was an affine function. In the general case F could be a nonlinear map, even though its image is a linear code. This however cannot happen with binary codes of dimension 2 in which the evaluation vectors of the functions $f_1,\dots,f_n$ have even weight.

Definition 10. A code which is not equivalent to any linear code is called a *strictly nonlinear* code.

Proposition 11. *The evaluation vectors of the non-constant components of a linear code are balanced, namely each $\alpha\in\mathbb{F}_q$ appears $\frac{M}{q}$ times. Moreover M has to be a power of q.*

When considering linear codes, there is no need to consider both the weight distribution and the distance distribution. The first important property of the weight distribution is that $A_0=1$ for any linear

code. This follows directly from Definition 9. Moreover, the minimum distance of a linear code coincides with the minimum of the weight distribution. The proof relies again on the definition itself of linear code: if c_1 and c_2 are two codewords at minimum distance, their difference is a codeword at minimum distance from the zero codeword, namely a codeword of minimum weight.

Since a linear code C is a vector subspace of $(\mathbb{F}_q)^n$ we can consider its dual, which is the vector subspace of $(\mathbb{F}_q)^n$ whose elements are orthogonal to C. The dual code of C coincides with the kernel of any its generator matrix, hence its dimension is $n - \dim(C)$.

Example 2. Let C be the binary linear code generated by the matrix

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

as in Example 1. The kernel of G is the vector subspace $\langle (1,1,1) \rangle$, meaning that the dual of C is the code generated by the matrix

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

Definition 12. The *dual* of a linear code C is the linear code whose codewords are orthogonal to C. The dual of C is denoted as $C^\perp$ and its generator matrix is usually denoted as H. H is called the *parity-check matrix* of C,

By using the parity-check matrix, it is possible to prove several interesting properties of linear codes. Other than the following known result, we will use parity-check matrices also to show the link between puncturing and shortening of codes in Section 1.2.

Proposition 13. *Let C be an $[n,k]_q$ linear code and let H be its parity-check matrix. Then $\mathrm{d}(C) = d$ if and only if the minimum number of linear dependent columns of H is d.*

Proof. If $c \in C$ is a codeword of minimum weight d, then $H \cdot c^t = 0$. This means that there is a linear relation between the columns of H indexed by $\mathrm{supp}(c)$. □

For notation purposes, we will often use the following non standard definition.

Definition 14. Let C be an $(n, M)_q$ code. We call *codebook* of C any $M \times n$ table whose M rows are the codewords of C. With a slight abuse of notation, we will often denote a codebook of C with the symbol C itself.

Codebooks are usually associated to nonlinear codes for encoding procedures. Indeed, we remark that the choice of a codebook for C is equivalent to order its codewords. Clearly, permutations of the rows of a codebook lead to other codebooks of the same code, in the same manner in which there exist several generator functions for the same code. Field permutations applied to the columns of C, permutations of its columns and translations of its rows by a single fixed vector of $(\mathbb{F}_q)^n$ lead to equivalent codes.
Once a codebook has been chosen we have a table

$$C = \{c_{i,j}\}_{i=0,\dots,M-1,\ j=1,\dots,n},$$

so we will denote with c_i the i-th codeword of C, which is the i-th row of its codebook, and with $c_{i,j}$ its j-th coordinate.

In the next section we present systematic codes, a family of codes containing linear codes.

1.1 Definition of systematic codes

Systematic codes form an important family of nonlinear codes. As we will show in Chapter 5, in the general case systematic codes can achieve better error correction capability than any linear code with the same parameters. On the other hand, due to their particular structure, systematic codes can achieve faster encoding and decoding procedures than nonlinear non-systematic codes. Moreover, many known families of optimal codes are systematic codes (see e.g. [26], [40]).

Definition 15. An $(n, q^k, d)_q$ systematic code C is the image of an injective map $F : (\mathbb{F}_q)^k \to (\mathbb{F}_q)^n$, $n \geq k$, s.t. a vector $X = (x_1, \ldots, x_k) \in (\mathbb{F}_q)^k$ is mapped to a vector

$$(x_1, \ldots, x_k, f_{k+1}(X), \ldots, f_n(X)) \in (\mathbb{F}_q)^n,$$

where the f_i are maps from $(\mathbb{F}_q)^k$ to $\mathbb{F}_q$. The coordinates from 1 to k are called *systematic*, while those from $k+1$ to n are called *non-systematic*.

A more general statement defines systematic codes as nonlinear codes equivalent to those described in Definition 15.

Definition 16. Let C be an $(n, q^k, d)_q$ code, and let $F = (f_1, \ldots, f_n)$ be a generator function of C. C is a *systematic code* if there exist k indices $i_1, \ldots, i_k$ for which the function $(f_{i_1}, \ldots, f_{i_k})$ is a permutation of $(\mathbb{F}_q)^k$.

According to Definition 15, any linear code is equivalent to a systematic code. The proof involves permutation of the columns of its generator matrix and application of the Gauss reduction to its rows. In this way we can find an equivalent code for which the generator matrix G is in the form

$$G = \left[\begin{array}{c|c} I_k & R_{n-k} \end{array} \right],$$

where I_k is the identity of order k and R_{n-k} is a $k \times (n-k)$ matrix. By applying instead Definition 16, each linear code can directly be called a systematic code.

Proposition 17. *Let C be a systematic code as in Definition 15. Then C is a linear code if and only if the components $f_{k+1}, \ldots, f_n$ are linear functions.*

Proof. If $f_{k+1}, \ldots, f_n$ are linear maps and c_1, c_2 are in C, then both αc_1 and $c_1 + c_2$ are in C, hence C is linear.

On the other hand, if a component f_j is not linear, then there exist two vectors $a = (a_1, \ldots, a_k)$ and $b = (b_1, \ldots, b_k)$ for which $f_j(a) + f_j(b) \neq f_j(a + b)$ and this implies that C is not a linear code. □

1.2 Puncturing, shortening and extending

The three most simple and common ways of obtaining new codes starting from known ones are

- puncturing,
- shortening and
- extending.

In this section we briefly describe these methods and we discuss some of the properties of the obtained code with respect to the parameters of the known code.

Definition 18. Puncturing a code C in the j-th coordinate means deleting from all codewords of C the j-th coordinate. We denote with $\dot{C}_{j_1,\ldots,j_s}$ the code obtained by puncturing C in the coordinates $j_1, \ldots, j_s$, or simply with $\dot{C}$ whenever we do not need to specify which coordinates were punctured.

The following proposition links the generator functions of $\dot{C}_j$ to the generator function of C, and follows directly from Definition 18.

Proposition 19. *Let $F = (f_1, \ldots, f_{j-1}, f_j, f_{j+1}, \ldots, f_n)$ be a generator function of a code C.*
Then $\dot{F}_j = (f_1, \ldots, f_{j-1}, f_{j+1}, \ldots, f_n)$ is a generator function for $\dot{C}_j$.

Given two codewords $c_1,\ c_2 \in C$ and the corresponding codewords $\dot{c}_1$ and $\dot{c}_2$ in $\dot{C}_j$, we have

$$\begin{aligned} \mathrm{d}(\dot{c}_1, \dot{c}_2) &= \mathrm{d}(c_1, c_2) && \text{if } c_{1,j} = c_{2,j} \\ \mathrm{d}(\dot{c}_1, \dot{c}_2) &= \mathrm{d}(c_1, c_2) - 1 && \text{if } c_{1,j} \neq c_{2,j} \end{aligned}$$

As a consequence, by puncturing C in a single coordinate, we find $\dot{C}$ whose distance is either equal to $\mathrm{d}(C)$ or to $\mathrm{d}(C) - 1$. In particular, $\mathrm{d}(\dot{C}_j) = \mathrm{d}(C)$ if and only if the component f_j is a constant function.

Let us consider now two codewords c_1 and c_2 in C at minimum distance d, and let $\mathrm{supp}(c_1 - c_2) = \{j_1, \ldots, j_d\}$. The puncturing of C in the coordinates $j_1, \ldots, j_d$ is a code with strictly less than M codewords. The proof of this is a straightforward consequence of the fact that both c_1 and c_2 are equal in all coordinates non indexed by elements in $\mathrm{supp}(c_1 - c_2)$, namely in $\dot{C}_{j_1,\ldots,j_d}$ we have $\dot{c}_1 = \dot{c}_2$. Let C be an $[n, k, d]_q$ linear code, and $c, \bar{c} \in C$ with $\bar{c}$ of minimum weight d. From the linearity of C it holds that $c + \alpha\bar{c} \in C$ for each $\alpha \in \mathbb{F}_q$, and these q codewords become the same after puncturing C in the coordinates $\mathrm{supp}(\bar{c})$. It follows that, by puncturing a linear code C in the support of a codeword of minimum weight, we obtain a new linear code whose dimension is strictly less than $\dim(C)$.

Definition 20. Let C be an $(n, M, d)_q$ code. *Shortening* C in the j-th coordinate means to consider the subset S of C containing all codewords whose j-th coordinate is equal zero, and then puncturing S in the j-th coordinate.
We denote with $\ddot{C}_{j_1,\ldots,j_s}$ the code obtained by shortening C in the coordinates $j_1, \ldots, j_s$, or simply with $\dot{C}$ whenever we do not need to specify which coordinates were shortened.

Similar to puncturing, shortening a code C produces a new code whose length is $\mathrm{len}(C) - 1$. On the other hand, the definition of shortening implies that the distance of $\ddot{C}$ is at least equal to the distance of C. Any pair of codewords in $\ddot{C}_j$ correspond to a pair of codewords in C whose j-th coordinates are equal, hence the shortening cannot have decreased their mutual distance.
Regarding the size of $\ddot{C}$, things are more complicated than with puncturing. First, if the j-th component of C is the constant 0, then shortening and puncturing C in such coordinate lead to the same

code. On the other hand, if the j-th coordinate is different from 0 for all codewords, then $\ddot{C}_j = \emptyset$.
In the general case, we have the following result.

Proposition 21. *Let C be an $(n, M)_q$ code with distance larger than 1, and let θ_α be a permutation of $\mathbb{F}_q$ which map α to 0.*
We consider the set $\{C_\alpha\}$ of equivalent codes obtained from C by applying θ_α to the j-th coordinate of C for each $\alpha \in \mathbb{F}_q$. We consider the set of codes $S = \left\{\ddot{C}_\alpha\right\}_{\alpha\in\mathbb{F}_q}$ obtained by shortening the codes $\{C_\alpha\}$ in the j-th coordinate. Then

1. *the codes in S are disjoint, meaning that a codeword cannot belong to more than one code in the set.*
2. *The union of all codes in S is the code $\dot{C}_j$.*
3. *There is at least an $\alpha \in \mathbb{F}_q$ such that $\left|\ddot{C}_\alpha\right| \geq \frac{M}{q}$.*

Proof. Without loss of generality, let us assume that $j = n$.

1. If a codeword $\ddot{c}$ belongs to more than a single code in S, then the concatenation $(\ddot{c}|0)$ belongs to more than a code in $\{C_\alpha\}$, say C_α and C_β. This means that both $(\ddot{c}|\alpha)$ and $(\ddot{c}|\beta)$ are in C, hence the minimum distance of C is 1.
2. If $\ddot{c}$ is in $\ddot{C}_\alpha$, then $(\ddot{c}|\alpha) \in C$. Moreover, if $(\ddot{c}|\alpha)$ is a codeword of C, then $\ddot{c} \in \ddot{C}_\alpha$. This implies that by joining the codes in S we obtain the set of all codewords in C to which we have removed the last coordinate, and this is the definition of $\dot{C}$.
3. Follows directly from $|C| = M$ and points 1. and 2. of this proposition.

□

If we consider linear codes we have a deeper characterisation given by the following proposition.

Proposition 22. *Let C be an $[n, k, d]_q$ linear code and let $D = C^\perp$ the dual of C. Then*

1. $\mathrm{len}(\ddot{C}) = n - 1$;
2. $\mathrm{d}(\ddot{C}) \geq d$;
3. *if the j-th component of C is the constant 0, then $\ddot{C}_j = \dot{C}_j$. In particular the dimension of $\ddot{C}$ is k.*
4. *if the j-th component of C is not constant, then the dimension of $\ddot{C}_j$ is $k - 1$.*
5. $\ddot{C}_j = \left(\dot{D}_j\right)^\perp$.

Proof. 1. It follows directly from the definition of shortening.

2. The codewords in $\ddot{C}$ are obtained by puncturing a subset of C containing elements whose j-th coordinate is equal 0. Since this set is contained in C, its codewords are at distance at least d, hence by deleting a coordinate which is constantly 0 their pairwise distances cannot decrease.

3. $\ddot{C}_j$ is by definition the puncturing of the subcode of C consisting of codewords whose j-th coordinate is 0. If all codewords in C possess this property, then shortening and puncturing lead to the same code. Moreover, the size of $\ddot{C}$ is equal to the size of C.

4. Since the j-th component of C is linear, then its evaluation vector is balanced, namely each $\alpha \in \mathbb{F}_q$ appears the same amount of times, i.e. $\frac{|C|}{q} = q^{k-1}$. This holds in particular for $\alpha = 0$, hence $|\ddot{C}_j| = q^{k-1}$.

5. D contains all vectors in $(\mathbb{F}_q)^n$ which are orthogonal to the codewords in C. Let us consider a codeword $c \in C$ whose j-th coordinate is 0 and a codeword $\bar{c}$ in D. The scalar product $c \cdot \bar{c} = 0$

does not depend on the j-th coordinate of $\bar{c}$, hence by puncturing both c and $\bar{c}$ in the j-th coordinate we still obtain a pair of orthogonal vectors. This imply that by puncturing in the j-th coordinate both the set of codewords in C whose j-th coordinate is 0 and D we obtain two orthogonal codes. To prove that these two codes, namely $\ddot{C}_j$ and $\dot{D}_j$, are dual codes we observe that $\dim \ddot{C}_j = \dim \left(\dot{D}_j\right)^{\perp}$.

□

Proposition 23. *Let C be an $(n, q^k, d)_q$ systematic code, and C' be the code obtained by shortening C in a systematic coordinate. Then C' is an $(n-1, q^{k-1}, d')_q$ systematic code with $d' \geq d$.*

Proof. To obtain C', consider the code

$$C'' = \left\{ F(X) \mid X = (0, x_2, \ldots, x_k) \in (\mathbb{F}_q)^k \right\},$$

i.e. the subcode of C which is the image of the set of messages whose first coordinate is equal to 0. Then C'' is such that $\dim(C'') = k-1$ and $\mathrm{d}(C'') \geq d$. Since, by construction, all codewords have the first coordinate equal to zero, we obtain the code C' by puncturing C'' on the first coordinate, so that

$$\begin{cases} \mathrm{len}(C') = n-1 \\ d' = \mathrm{d}(C') = \mathrm{d}(C'') \geq d. \end{cases}$$

□

Lemma 24. *For any $(n, q^k, d)_q$ systematic code C, there exists an $(n, q^k, \bar{d})_q$ systematic code $\bar{C}$ for any $1 \leq \bar{d} \leq d$.*

Proof. Since $n > k$, we can consider the code C^1 obtained by puncturing C in a non-systematic coordinate. C^1 is an $(n-1, q^k, d^{(1)})_q$ systematic code. Of course, either $d^{(1)} = d$ or $d^{(1)} = d-1$.
By puncturing at most $n-k$ non-systematic coordinates, we will find a code whose distance is 1. Then there must exists an $i \leq n-k$ such

that the code C^i, obtained by puncturing C in the last i coordinates, has distance equal to $\bar{d}$. Once the $(n-i, q^k, \bar{d})$ code C^i has been found, we can obtain the claimed code $\bar{C}$ by padding i zeros to all codewords in C^i. □

The proofs of Proposition 23 and Lemma 24 rely on the systematic property of C, hence they hold for linear codes as well. For non-systematic codes we have the following more general statement.

Lemma 25. *For any $(n, M, d)_q$ code C, there exists an $(n, M, \bar{d})_q$ code $\bar{C}$ for any $1 \leq \bar{d} \leq d$.*

Proof. By proceeding in a similar manner as in the proof of Lemma 24, we puncture C till we reach a code whose distance is $\bar{d}$. Then $\bar{C}$ is obtained by padding the necessary amount of zeros to all codewords of the punctured code. □

We conclude this section with the notion of extension of a code. The idea is to add a new coordinate to a code C, hopefully increasing its distance. The definition of the new component is given in terms of the components of C, so in general it is a map $(\mathbb{F}_q)^n \to \mathbb{F}_q$ evaluated at the components of C. The result is therefore a map $(\mathbb{F}_q)^k \to \mathbb{F}_q$ which is used as the new component.
By the definition above, there are several possible ways to define an extension of a code - as much as the maps $(\mathbb{F}_q)^k \to \mathbb{F}_q$ - however the new-added component is usually defined by

$$f_{n+1} = -\sum_j f_j. \tag{1.1}$$

In this work we will always refer to this particular type of extension as the extension of C.

Definition 26. Let C be a code with length n, let $f_1, \ldots, f_n$ be its components and let f_{n+1} be as in equation (1.1). We denote with $E(C)$ the extension of C defined by the components

$$(f_1, \ldots, f_n, f_{n+1}).$$

Notice that f_{n+1} is the composition between a linear function and the components of C. This lead to the following results.

Proposition 27. *$E(C)$ is linear if and only if C is a linear code.*

Proof. If $E(C)$ is linear then its components are linear. In particular $f_1, \ldots, f_n$ are linear functions, so the components of C are all linear functions. For the other way around, if $f_1, \ldots, f_n$ are linear, then also $f_{n+1} = -\sum_{j=1}^{n} f_j$ is a linear function, making $E(C)$ a linear code. □

Proposition 28. *The distance of $E(C)$ is either equal to $\mathrm{d}(C)$ or $\mathrm{d}(C) + 1$.*

Proof. The distance of C cannot decrease after an extension, and by adding a single coordinate it cannot increase by more than 1. □

Proposition 29. *The sum of the coordinates of each codeword in $E(C)$ is 0.*

Proof. It follows directly from the definition of $E(C)$. □

Proposition 29 lead to very interesting properties, mainly in the case of binary codes, as stated by the following results.

Corollary 30. *Let C be a binary code. Then $E(C)$ contains only codewords of even weight.*

Proof. It follows from Proposition 29. A binary codeword has even weight if and only if the sum of its coordinates is 0. □

Proposition 31. *Let C be an $(n, M, d)_2$ code with odd distance d. Then $E(C)$ has distance equal to $d + 1$.*

Proof. By Corollary 30 in $E(C)$ there are only codewords with even weight, and the distance between two such codeword cannot be odd. By Proposition 28 the distance of $E(C)$ is either d or $d + 1$, hence it has to be equal to $d + 1$ since d is odd. □

In the general case, extending and puncturing do not commute, i.e. while it is true that by extending and then puncturing a code C we obtain again C, it is not true that extending a punctured code will lead to the original code. In the following proposition we assume without loss of generality that both extending and puncturing are referred to the last coordinate of the code.

Proposition 32. *Let C be a code. Then*

1. $\dot{E}(C) = C$,
2. $E(\dot{C}) \neq C$, *and*
3. *in the linear binary case $E(\dot{C}) = C$ if and only if C has only codewords of even weight.*

Proof. 1 and 2 directly follow from the definitions, while 3 follows from Corollary 30. □

Chapter 2

Bounds on code parameters

> *It may seem surprising that we should define a definite capacity C for a noisy channel since we can never send certain information in such a case. [...] Nature takes payment by requiring just that much uncertainty, so that we are not actually getting any more than C through correctly.*

Claude Shannon

A classical problem in coding theory is to determine the parameters of optimal codes, and this characterisation is usually carried on by presenting bounds on the minimum distance, on the size, or on the length of codes. Since two equivalent codes have the same parameters, we can always assume that the zero codeword belongs to C.
In this chapter we present an overview of the most common bounds on the size of a code. Most bounds estimate the maximum number of codewords in a code whose length and distance are known. This is not the case for the Griesmer bound, whose classical description gives a lower-bound on the length of a linear code with given size and minimum distance. We will discuss extensively the Griesmer bound in Chapter 5.

Definition 33. We denote with

- $A_q(n, d)$ the maximum size of a nonlinear code whose length is n and whose distance is d.
- $B_q(n, d)$ the maximum size of a linear code whose length is n and whose distance is d.
- $N_q(M, d)$ the minimum length of a nonlinear code whose size is M and whose minimum distance is d.
- $S_q(k, d)$ the minimum length of a nonlinear systematic code with dimension k and minimum distance d.
- $L_q(k, d)$ the minimum length of a linear code with dimension k and minimum distance d.

Clearly there exist some relations between these values, as explained by the following proposition.

Proposition 34. *$B_q(n, d) \leq A_q(n, d)$ and $N_q(M, d) \leq S_q(k, d) \leq L_q(k, d)$.*

Proof. It follows directly from Definition 33. □

Definition 35. We denote with $S_r(x)$ the Hamming sphere centred at x of radius r, and with $\mathrm{Vol}_q(n,r) = \sum_{i=0}^{r} \binom{n}{i}(q-1)^i$ the volume of a sphere in $(\mathbb{F}_q)^n$ of radius r.

2.1 Sphere packing bound

The sphere packing bound, also known as Hamming bound, is a first and very important bound on the size of a code and was proposed in 1950 by R. W. Hamming [19]. Codes meeting the sphere packing bound are called *perfect codes*. In a perfect code the spheres of radius $t = \left\lfloor \frac{n-1}{2} \right\rfloor$ centred at codewords cover the entire space $(\mathbb{F}_q)^n$, or equivalently for each vector v in the space the sphere or radius t contains a codeword. This imply that each vector can be successfully decoded without ambiguity.

Theorem 36 (Sphere packing bound)**.**

$$B_q(n,d) \leq A_q(n,d) \leq \frac{q^n}{\mathrm{Vol}_q(n,t)},$$

where $t = \left\lfloor \frac{d-1}{2} \right\rfloor$.

Proof. Given an $(n,M,d)_q$ code, the M spheres of radius t centred at the M codewords are pairwise disjoint. □

2.2 Gilbert bound

The following result is a lower bound on the size of a code proposed in 1952 by E. N. Gilbert [12]. The covering radius $\rho(C)$ of an $(n,M,d)_q$ code is the minimum integer r such that the union of the spheres of radius r centred at the M codewords is equal to the entire space $(\mathbb{F}_q)^n$. An argument similar to the proof of the Sphere packing bound leads to the following lower bound for $B_q(n,d)$.

Theorem 37 (Gilbert bound)**.**

$$A_q(n,d) \geq B_q(n,d) \geq \frac{q^n}{\mathrm{Vol}_q(n,d-1)}$$

Proof. The covering radius of an $[n, B_q(n,d), d]_q$ linear code C is at most $d-1$. The proof of this fact is by contradiction. Suppose there exists an element of $(\mathbb{F}_q)^n$ at distance at least d from any codeword in C. In this case we could construct another linear code $\bar{C}$ with the same length and distance as C, and dimension strictly larger than $B_q(n,d)$, which leads to the claimed contradiction.
Since the spheres of radius $d-1$ centred at the codewords of C cover $(\mathbb{F}_q)^n$, it follows that $B_q(n,d)$ is at least equal to the number of such spheres, and the result follows. □

2.3 Varshamov bound

Another lower bound for $B_q(n,d)$ was proposed in 1957 by R. R. Varshamov [50].

Lemma 38. *Let n, k and d be integers such that $2 \leq d \leq n$ and $1 \leq k \leq n$, and let q be a prime power. If*

$$\sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i < q^{n-k}, \tag{2.1}$$

then there exists an $n-k \times n$ matrix H over $\mathbb{F}_q$ such that every $d-1$ columns of H are linearly independent.

Proof. We use the following algorithm to find the n columns $h_1, \ldots, h_n$ of H. Notice that each column is a vector in $(\mathbb{F}_q)^{n-k}$. Choose:

- h_1 to be any nonzero vector;
- h_2 to be any vector which is not a multiple of h_1;
- h_j to be any vector which is not a linear combination of $d-2$ (or fewer) of the vectors $h_1, \ldots, h_{j-1}$, for each $j \leq n$.

If we can carry out this algorithm to completion, then $h_1, \ldots, h_n$ are the columns of an $n-k$ by n matrix with the wanted property.
Suppose we have found $h_1, \ldots, h_j$, with $1 \leq j \leq n$. The number of linear combinations of $d-2$ or fewer of $h_1, \ldots, h_j$ is

$$\sum_{i=0}^{d-2} \binom{j}{i} (q-1)^i \leq \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i.$$

Hence if equation (2.1) holds, then there is some vector h_{j+1} which is not a linear combination of $d-2$ (or fewer) of $h_1, \ldots, h_j$. By induction we conclude. □

H can be considered as the parity check matrix of a linear code C of length n and dimension at least k. Since every $d-1$ columns of H are linear independent, C has distance at least d by Proposition 13.

Corollary 39. *Let n, k and d be integers such that $2 \leq d \leq n$ and $1 \leq k \leq n$. Then there exists an $[n,k]_q$ linear code with minimum distance at least d, provided that*

$$1 + \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i \leq q^{n-k}. \tag{2.2}$$

Theorem 40 (Varshamov bound)**.**

$$B_q(n,d) \geq q^{n - \lceil \log_q \left(1 + \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i\right) \rceil}.$$

Proof. The largest integer k satisfying equation (2.2) is equal to

$$\left\lfloor n - \log_q \left(1 + \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i\right) \right\rfloor,$$

hence the theorem follows from Corollary 39. □

2.4 Singleton bound

One of the most commonly used bounds was presented in 1964 by R. Singleton [44]. Codes meeting the Singleton bound are known as

Maximum Distance Separable (MDS) codes, and are of great importance in both Coding Theory and Cryptography.

Theorem 41 (Singleton bound)**.**

$$A_q(n,d) \leq q^{n-d+1}.$$

Proof. By puncturing an $(n,M,d)_q$ code in $d-1$ coordinates, we obtain an $(n,M)_q$ code with distance at least 1, hence M is at most q^{n-d+1}. □

By applying the Singleton bound to linear codes we obtain the following corollary.

Corollary 42 (Singleton bound for linear codes)**.** *Let C be an $[n,k,d]_q$ linear code. Then*

$$k \leq n-d+1.$$

Let consider now an $[n,k,d]_q$ code C. If D is any subcode of C, we define the support $\mathrm{supp}(D)$ of D as the set of coordinates where not all the codewords of D are zero. Equivalently, the support of D is the set of indices corresponding to the nonzero columns of the generator matrix of D.
For $1 \leq r \leq k$, the r-th-generalized Hamming Weight of C is defined as

$$d_r(C) = d_r = \min\left\{|\mathrm{supp}(D)| \mid D \text{ is an } [n,r] \text{ subcode of } C\right\}.$$

Notice that

$$d = d_1 < d_2 < \ldots < d_k \leq n,$$

and the set $\{d_1,\ldots,d_k\}$ is called the *weight hierarchy* of C. The weight hierarchy is invariant under any monomial map applied to C.

Theorem 43 (Generalized Singleton bound)**.** *Let C be an $[n,k,d]_q$ linear code. For $1 \leq r \leq k$*

$$d_r \leq n-k+r.$$

Proof. It follows by induction on $k-r$, starting with $d_k \leq n$ and using $d_1 < \ldots < d_k$. □

2.5 Griesmer bound

The Griesmer bound, which can be seen as an extension of the Singleton bound [23, Section 2.4] in the linear case, was introduced by Griesmer [14] in the case of binary linear codes and then generalized by Solomon and Stiffler [45] in the case of q-ary linear codes. It is known that the Griesmer bound is not always sharp [34], [35], [49]. Important examples of linear codes meeting the Griesmer bound are the simplex code [23, Section 1.3] and the $[11, 5, 6]_3$ Golay code [13], [23, Section 1.12].

Definition 44. Let C be a linear code, and let $c \in C$. We denote with $\mathrm{Res}(C, c)$ the residual code of C with respect to c, namely the linear code obtained by puncturing C in the nonzero coordinates of c.

Lemma 45. *If C is an $[n, k, d]_q$ code and $c \in C$ has weight d, then $\mathrm{Res}(C, c)$ is an $[n-d, k-1, \dot{d}]_q$ code where $\dot{d} \geq \left\lceil \frac{d}{q} \right\rceil$.*

Proof. By replacing C with an equivalent code, we can assume $c = (1, \ldots, 1, 0, \ldots, 0)$, namely, the nonzero coordinates of c are the first d coordinates.
First, we prove that the dimension of $\mathrm{Res}(C, c)$ is $k-1$. Assume that this is not true, which means that the dimension is strictly less than $k-1$. Then there exists a nonzero codeword $x \in C$ of the form $x = (x_1, \ldots, x_d, 0, \ldots, 0)$, and x is not a multiple of c. Observe that $x_1, \ldots, x_d$ are nonzero, otherwise the weight of x would be strictly less than d. We consider now the codeword $x_1 \cdot c = (x_1, \ldots, x_1, 0, \ldots, 0)$ which belongs to C due to the linearity of C. Then $d(x, x_1 \cdot c) < d$, and we have a contradiction. Hence the dimension of $\mathrm{Res}(C, c)$ is $k-1$.
We now provide a lower bound for $\dot{d}$. Consider a nonzero codeword $v \in$

$\mathrm{Res}(C, c)$, and let $u \in (\mathbb{F}_q)^d$ be such that the concatenation $(u|v) \in C$. Observe that there exists an $\alpha \in \mathbb{F}_q$ such that at least d/q coordinates of u equal α. Then

$$d \leq d((u|v), \alpha \cdot c) \leq d - \frac{d}{q} + w(v),$$

from which it follows that

$$w(v) \geq \frac{d}{q}.$$

Since this is true for all codewords in $\mathrm{Res}(C, c)$, then $\dot{d} \geq \frac{d}{q}$. □

Theorem 46 (Griesmer bound)**.** *Let $k \geq 1$. Then*

$$L_q(k, d) \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

Proof. Consider an $[n, k, d]_q$ code. The proof is by induction on k. If $k = 1$ the conclusion holds. Assume that $k > 1$ and let $c \in C$ be a codeword of weight d. By Lemma 45 the residual code $\mathrm{Res}(C, c)$ is an $[n - d, k - 1, \dot{d}]_q$ code with $\dot{d} \geq \left\lceil \frac{d}{q} \right\rceil$. Applying the inductive assumption, we have

$$n - d \geq \sum_{i=0}^{k-2} \left\lceil \frac{d}{q^{i+1}} \right\rceil.$$

□

2.6 Plotkin bound

The following is an upper bound due to M. Plotkin, first present in 1960 [39]. Binary codes meeting the Plotkin bound are of great importance for several reasons. First, it seems that for each (n, d) pair there exists a code whose size meets the bound. Furthermore, optimal codes in the Plotkin range are equidistant codes, i.e. each pair of codewords is at distance exactly d. Finally, as we will see in Chapter 3, these codes have deep connections to Hadamard matrices.

Theorem 47 (Plotkin bound)**.** *Let C be an $(n, M, d)_q$ code such that $rn < d$, with $r = 1 - q^{-1}$. Then*

$$M \leq \left\lfloor \frac{d}{d - rn} \right\rfloor$$

Proof. Consider $S = \sum_{x \in C} \sum_{y \in C} d(x, y)$. Since $d(x, x) = 0$, while $d(x, y) \geq d$ for each pair $x \neq y$, it follows that

$$M(M-1)d \leq S. \tag{2.3}$$

Let A be a codebook for C. For $1 \leq i \leq n$, let $n_{i,\alpha}$ be the number of times $\alpha \in \mathbb{F}_q$ occurs in the i-th column of A. We have

$$S = \sum_{i=1}^{n} \sum_{\alpha} n_{i,\alpha} \left(M - n_{i,\alpha}\right) = nM^2 - \sum_{i=1}^{n} \sum_{\alpha} n_{i,\alpha}^2.$$

Using the Cauchy-Schwartz inequality, we observe that

$$\sum_{\alpha} n_{i,\alpha}^2 = q^{-1} \left(\sum_{\alpha} n_{i,\alpha}^2 \right) \left(\sum_{\alpha} 1 \right) \geq q^{-1} \left(\sum_{\alpha} n_{i,\alpha} \right)^2,$$

hence

$$S \leq nM^2 - \sum_{i=1}^{n} q^{-1} \left(\sum_{\alpha} n_{i,\alpha} \right)^2.$$

Since $\sum_{\alpha} n_{i,\alpha} = M$, we have

$$S \leq nM^2 - nq^{-1}M^2 = nM^2 \left(1 - q^{-1}\right) = nrM^2. \tag{2.4}$$

Combining equations (2.3) and (2.4) we obtain

$$M(M-1)d \leq nrM^2,$$

from which we have

$$M(d - rn) \leq d.$$

□

Theorem 48 (Binary Plotkin Bound)**.** *For a binary code the Plotkin bound can be specialised in the following way.*

- *If d is even and $n < 2d$, $A_2(n,d) \leq 2\left\lfloor \frac{d}{2d-n} \right\rfloor$.*
- *If d is even, $A_2(2d,d) = 4d$.*
- *if d is odd and $n < 2d+1$, $A_2(n,d) \leq 2\left\lfloor \frac{d+1}{2d+1-n} \right\rfloor$.*
- *If d is odd, $A_2(2d+1,d) = 4d+4$.*

2.7 Johnson bounds

Another remarkable result is due to S. Johnson [24]. In 1962 he presented a new upper bound by examining binary constant-weight codes, namely codes in which all codewords have the same Hamming weight. Define $A_2(n,d,w)$ to be the maximum number of binary codewords of length n and weight w which are distant at least d apart. In the following we consider upper bounds on $A_2(n,d,w)$ and $A_2(n,d)$ due to Johnson.

Theorem 49. *Let C be an $(n,M)_2$ constant-weight binary code with codewords of weight w. Suppose that every pair of distinct codewords have at most λ ones in common positions. If $w^2 > n\lambda$, then*

$$M \leq \left\lfloor \frac{n(w-\lambda)}{w^2 - n\lambda} \right\rfloor.$$

Proof. Let $A = [c_{i,j}]$ be the codebook of C. We prove the bound by estimating the sum

$$S = \sum_{i=1}^{M} \sum_{\substack{j=1,\\ j\neq i}}^{M} \sum_{k=1}^{n} c_{i,k}c_{j,k}.$$

By hypothesis,

$$S \leq M(M-1)\lambda. \tag{2.5}$$

For $1 \leq k \leq n$, let m_k be the number of ones in the k-th column of A. Then

$$S = \sum_{k=1}^{n} m_k(m_k - 1)$$

and since $\sum_{k=1}^{n} m_k = wM$, it follows that

$$S = \sum_{k=1}^{n} m_k^2 - wM.$$

By Cauchy-Schwartz inequality we have that

$$\sum_{k=1}^{n} m_k^2 \geq n^{-1} \left(\sum_{k=1}^{n} m_k \right)^2,$$

hence

$$S \geq n^{-1} \left(\sum_{k=1}^{n} m_k \right)^2 - wM = n^{-1}(wM)^2 - wM. \tag{2.6}$$

Equations (2.5) and (2.6) lead to

$$w^2M^2 - wMn \leq M^2\lambda n - M\lambda n,$$

and the theorem follows. □

Corollary 50 (Restricted Johnson bound for $A_2(n, d, w)$)**.**

$$A_2(n, 2e-1, w) = A_2(n, 2e, w) \leq \left\lfloor \frac{ne}{w^2 - nw + ne} \right\rfloor,$$

provided that $w^2 - nw + ne > 0$.

Proof. The corollary follows from Theorem 49: if C is an $(n, M, 2e)_2$ constant-weight code with codewords of weight w, then every pair of distinct codewords have at most $w - e$ ones in common positions. Moreover, since any pair of binary vectors of the same weight are an even distance apart, $A_2(n, 2e-1, w) = A_2(n, 2e, w)$.
Using Theorem 49 with $\lambda = w - e$, the claim follows. □

The name restricted Johnson bound for $A_2(n, d, w)$ comes from the fact that the hypothesis $w^2 - nw + ne > 0$ is necessary. Another bound due to Johnson remove this hypothesis.

Theorem 51 (Unrestricted Johnson bound for $A_2(n, d, w)$)**.**

1. *if* $w < e$, *then* $A_2(n, 2e-1, w) = A_2(n, 2e, w) = 1$.

2. *if* $w \geq e$, *then* $A_2(n, 2e-1, w) = A_2(n, 2e, w)'$, *and*

$$A_2(n, 2e, w) \leq \left\lfloor \frac{n}{w} \left\lfloor \frac{n-1}{w-1} \left\lfloor \cdots \left\lfloor \frac{n-w+e}{e} \right\rfloor \cdots \right\rfloor \right\rfloor \right\rfloor .$$

Proof. Part 1. comes from the fact that two codewords of weight w are at most at distance $2w$ apart.
For part 2. we consider a binary code C of length n and distance at least $2e$ having $M = A_2(n, 2e, w)$ codewords each of weight w. Similarly to the proof of Theorem 49 we count the number $s = \sum_{k=1}^{n} m_k$ of ones in the codebook A of C.
Since C is a constant-weight code we obtain

$$s = wM = wA_2(n, 2e, w). \tag{2.7}$$

On the other hand we observe that each column of A contains at most $A_2(n-1, 2e, w-1)$ ones. In fact, by considering the $\dot{M} \times (n-1)$ submatrix of A obtained by removing a column from A and keeping only the rows in which there is a 1 in the removed column, we find a table corresponding to a constant-weight code with length $n-1$ and distance $2e$, in which every codeword has weight $w-1$. Together with equation (2.7) we obtain

$$wA_2(n, 2e, w) = s \leq n \cdot A_2(n-1, 2e, w-1),$$

which leads us to

$$A_2(n, 2e, w) \leq \left\lfloor \frac{n}{w} A_2(n-1, 2e, w-1) \right\rfloor .$$

Part 2. follows by induction with part 1. being used as final step. □

Observe that $A_2(n+1, 2e) = A_2(n, 2e-1)$. Due to this, to obtain a bound for $A_2(n, d)$ it is sufficient to consider the case d odd.

Theorem 52 (Johnson bound for $A_2(n,d)$)**.** *Let $d = 2e+1$. Then*

$$A_2(n,d) \leq \frac{2^n}{\sum_{i=0}^{e}\binom{n}{k} + \frac{\binom{n}{e+1}-\binom{d}{e}A_2(n,d,d)}{\left\lfloor\frac{n}{e+1}\right\rfloor}} \tag{2.8}$$

and

$$A_2(n,d) \leq \frac{2^n}{\sum_{i=0}^{e}\binom{n}{k} + \frac{\binom{n}{e}\left(\frac{n-e}{e+1}-\left\lfloor\frac{n-e}{e+1}\right\rfloor\right)}{\left\lfloor\frac{n}{e+1}\right\rfloor}}. \tag{2.9}$$

Proof. The second bound is implied by the first one and by Theorem 51, in fact

$$\binom{d}{e}A_2(n,d,d) \leq \binom{n}{e}\left\lfloor\frac{n-e}{e+1}\right\rfloor.$$

To prove bound (2.8), let C be an $(n,M,d)_2$ code. Let $N = \{x \in (\mathbb{F}_2)^n \mid d(C,x) = e+1\}$. Since the spheres of radius e centred at codewords are disjoint,

$$M\sum_{i=0}^{e}\binom{n}{i} + |N| \leq 2^n. \tag{2.10}$$

To complete the proof we need a lower bound on $|N|$.
We denote with X the set of codeword-vector pairs at distance $e+1$, namely

$$X = \{(c,x) \in C \times N \mid d(c,x) = e+1\}.$$

In the following we will obtain lower and upper estimates for $|X|$. To obtain the lower bound, we fix $c \in C$, and, since we can replace C with its translate $c+C$, we assume $c = 0$. The distance d is equal to $2e+1$, hence all vectors $x \in (\mathbb{F}_2)^n$ of weight $e+1$ satisfy either $d(C,x) = e$ or $d(C,x) = e+1$. The vectors x of weight $e+1$ which have distance e from C must be at distance e from a codeword of weight $d = 2e+1$. Because C has at most $A_2(n,d,d)$ codewords of weight d, there are at most $\binom{d}{e}A_2(n,d,d)$ such vectors x. Thus, there are at least $\binom{n}{e+1} - \binom{d}{e}A_2(n,d,d)$ vectors $x \in N$ such that $(0,x) \in X$. Therefore

$$M\left(\binom{n}{e+1} - \binom{d}{e}A_2(n,d,d)\right) \leq |X|.$$

To obtain instead an upper bound on $|X|$, we fix $x \in N$, and by replacing C with its translate $x + C$ we assume $x = 0$. Two distinct codewords c_1 and c_2 both at distance $e+1$ from 0 must have disjoint supports, since $d(c_1, c_2) \geq 2e + 1$. Thus there are at most $\left\lfloor \frac{n}{e+1} \right\rfloor$ codewords at distance $e + 1$ from 0, and therefore

$$|X| \leq |N| \left\lfloor \frac{n}{e+1} \right\rfloor .$$

Combining the obtained lower and upper bound on $|X|$ we obtain

$$M \frac{\binom{n}{e+1} - \binom{d}{e} A_2(n, d, d)}{\left\lfloor \frac{n}{e+1} \right\rfloor} \leq |N|.$$

By replacing $|N|$ in equation (2.10) with its lower bound, we obtain (2.8). □

2.8 Elias bound

The following bound was first presented by Elias, and nowadays it is mainly known as Elias-Bassalygo bound [5].

Lemma 53. *Let C be an $(n, M, d)_q$ code, and let $0 \leq e \leq n$ be a positive integer. Then there exists a Hamming ball of radius r with at least $\frac{M \mathrm{Vol}_q(n,r)}{q^n}$ codewords in it.*

Proof. Randomly pick an element y of $(\mathbb{F}_q)^n$. The expected size of overlapped region between C and the Hamming ball centred at y with radius r is

$$\mathrm{Vol}_q(n, r) \cdot \frac{M}{q^n}, \tag{2.11}$$

since y is randomly selected. Therefore there is at least one y such that the Hamming ball of radius r centred in y has size at least equal to (2.11), otherwise the expectation would have been smaller than this value. □

Lemma 54. *Let $\theta = \frac{q-1}{q}$, and let $r \leq \theta n$. Let C be an $(n, K, d)_q$ code such that its minimum weight is at most r. Then*

$$d \leq \frac{Kr}{K-1}\left(2 - \frac{r}{\theta n}\right).$$

Proof. Let $m_{i,\alpha}$ denote the number of occurrences of the symbol α in the i^{th} column of the codebook of C. We recall that we denote with $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ the multiplicative subgroup of $\mathbb{F}_q$. We know that

$$\sum_{\alpha \in \mathbb{F}_q} m_{i,\alpha} = K$$

and

$$\sum_{i=1}^{n} m_{i,0} = S \geq K(n-r),$$

because every row of the codebook has weight at most r. Therefore

$$\sum_{\alpha \in \mathbb{F}_q^*} m_{i,\alpha}^2 \geq (q-1)^{-1} \left(\sum_{\alpha \in \mathbb{F}_q^*} m_{i,\alpha} \right)^2 = (q-1)^{-1}(K - m_{i,0})^2$$

and

$$\sum_{i=1}^{n} m_{i,0}^2 \geq n^{-1} \left(\sum_{i=1}^{n} m_{i,0} \right)^2 = n^{-1} S^2.$$

As in the proof of the Plotkin bound, we compute the sum of the distances of all ordered pairs of rows of the codebook, obtaining that

$$\sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} m_{i,\alpha}(K - m_{i,\alpha}) \leq nK^2 - (q-1)^{-1}(qn^{-1}S^2 + nK^2 - 2KS).$$

In this inequality, we substitute $S \geq K(n-r)$, where we pick $r \leq \theta n$, and hence $S \geq q^{-1}nK$. We find

$$\sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} m_{i,\alpha}(K - m_{i,\alpha}) \leq K^2 r \left(2 - \left(\frac{r}{\theta n}\right)\right).$$

Since the number of pairs of rows is $K(K-1)$, we have

$$K(K-1)d \leq K^2 r \left(2 - \frac{r}{\theta n}\right),$$

which concludes the proof. □

Theorem 55 (Elias Bound)**.** *Let $\theta = 1 - q^{-1}$, let r be a positive integer such that $r \leq \theta n$ and $r^2 - 2\theta nr + \theta nd > 0$. Then*

$$A_q(n,d) \leq \frac{\theta nd}{r^2 - 2\theta nr + \theta nd} \cdot \frac{q^n}{\mathrm{Vol}_q(n,r)}.$$

Proof. Given an $(n, M, d)_q$ code C', consider the $(n, K, d)_q$ code C consisting of all the codewords in C' of weight at most r. Applying Lemma 53 we may assume that

$$K \geq \mathrm{Vol}_q(n,r) \cdot \frac{M}{q^n}.$$

We can now apply Lemma 54 to C. This yields

$$\mathrm{Vol}_q(n,r) \cdot \frac{M}{q^n} \leq \frac{\theta nd}{r^2 - 2\theta nr + \theta nd}.$$

□

Remark 56. Note that if $r = \theta n$ and $d > \theta n$, we obtain the Plotkin bound.

2.9 Zinoviev-Litsyn-Laihonen bound

In this section we show a bound first introduced by Zinoviev and Litsyn in 1984 [53] and then applied by Laihonen and Litsyn in 1998 [30]. The proof of the bound is based on the same ideas behind the Elias bound, hence we will make use of previous results instead of the entire original proof, which can be found in [30].

Theorem 57 (Zinoviev-Litsyn-Laihonen bound)**.** *Let $1 \leq d \leq n$, $d - 2r \leq n - t$, $0 \leq r \leq t$ and $0 \leq r \leq \frac{1}{2}d$. Then*

$$A_q(n,d) \leq \frac{q^t}{V_q(t,r)} A_q(n-t, d-2r).$$

Proof. Let C be an $(n, M, d)_q$ code with $M = A_q(n,d)$. Let us consider the code C_t obtained by puncturing C in t components, and

keeping only those codewords in which the chosen t components belong to a Hamming sphere of radius r. C_t is an $(n-t, M_t, d_t)_q$ code in which

$$M_t \geq \frac{M}{q^t} V_q(t,r) \qquad \text{and} \qquad d_t \geq d - 2r.$$

The fact that the length of C_t is $n-t$ comes directly from the puncturing of C in t positions, while $d_t \geq d-2r$ follows from the fact that the deleted parts of the selected codewords differ at most in $2r$ components.

To prove the bound on the number of codewords in C_t, we denote with $y_1, \ldots, y_M$ the M vectors in $(\mathbb{F}_q)^t$ corresponding to the punctured coordinates of C.

By Lemma 53, there exists a Hamming sphere $S_t(x)$ containing at least $\frac{M}{q^t} V_q(t,r)$ vectors among $y_1, \ldots, y_m$ and so the claim follows. □

2.10 Bellini-Guerrini-Sala bounds

In this last section we show two bounds by E. Bellini, G. Guerrini and M. Sala [7], which are specialized to systematic-emdedding codes. The first of these two bound is actually an improvement of the Zinoviev-Litsyn-Laihonen bound.

Definition 58. Let C be an $(n, M, d)_q$ code, and let $k = \lfloor \log_q(M) \rfloor$. We say that C is systematic embedding if C contains a systematic code D with $|D| = q^k$.

We remark that systematic-embedding codes are a generalization of systematic codes, and every systematic code C is systematic embedding with $D = C$. Several known families of maximal codes are either systematic or systematic-embedding codes (see e.g. [1], [26] and [40]).

Definition 59. We denote with $A_q^*(n,d)$ the maximum number of codewords that an $(n, M, d)_q$ systematic embedding code can contain.

We remark that $A_q^*(n,d) \leq A_q(n,d)$. To our knowledge there are no explicit counterexamples to $A_q^*(n,d) = A_q(n,d)$ prior to this work. We will discuss systematic codes in Chapter 5 and provide a proof that there exist values n and d for which $A_q^*(n,d) < A_q(n,d)$ by explicit computation in the case $n = 19$ and $d = 10$.

Proposition 60. *Let C be an $(n, M, d)_q$ code. Let $\varepsilon \geq 1$ be a positive integer such that for any $c \in C$ we have $\mathrm{w}(c) \geq d + \varepsilon$. Then*

$$M \leq A_q(n,d) - \frac{\mathrm{Vol}_q(n,\varepsilon)}{\mathrm{Vol}_q(n,d-1)}.$$

Proof. Let C be a code satisfying our hypothesis. C belongs to the set of all codes with distance d that are contained in $(\mathbb{F}_q)^n \setminus S_{d+\varepsilon-1}(0)$. Let D be any code of the largest size in this set, then $|C| \leq |D|$.
Clearly, any codeword c of D has weight $\mathrm{w}(c) \geq d + \varepsilon$. Consider also $\bar{D}$, the largest code in $(\mathbb{F}_q)^n$ of distance d and such that $D \subseteq \bar{D}$. By definition, the only codewords of $\bar{D}$ of weight greater than $d + \varepsilon - 1$ are those of D, while all other codewords of $\bar{D}$ are confined to the ball $S_{d+\varepsilon-1}(0)$. Thus:

$$M \leq |D| \leq |\bar{D}| \leq A_q(n,d)$$

and

$$\bar{D} \setminus D \subseteq S_{d+\varepsilon-1}(0).$$

Let $\rho = d-1$ and $r = d+\varepsilon-1$, so that $r-\rho = \varepsilon$, and let $N = \bar{D} \cap S_r(0)$. We have that $D = \bar{D} \setminus N$ and $|D| = |\bar{D}| - |N|$. By providing a lower bound on N we obtain an upper bound on $|D|$.
We start by proving $S_{r-\rho}(0) \subseteq \bigcup_{x \in N} S_\rho(x)$. Indeed, assume there exists a vector y not contained in the latter. Such vector is at distance at least $\rho + 1 = d$ from any codeword in N. Moreover, since $y \in S_{r-\rho}(0)$, its distance from $\bar{D} \setminus N$ is at least d. We can therefore obtain a new code $\bar{D} \cup \{y\}$ containing D and with distance d, contradicting the maximality of $\bar{D}$.

As a consequence, we have that

$$|N| \cdot \mathrm{Vol}_q(d, \rho) \geq \mathrm{Vol}_q(n, r - \rho),$$

which gives

$$|N| \geq \frac{\mathrm{Vol}_q(n, \varepsilon)}{\mathrm{Vol}_q(n, d-1)}.$$

Using previous observations, we obtain

$$M \leq |D| = |\bar{D}| - |N| \leq A_q(n, d) - \frac{\mathrm{Vol}_q(n, \varepsilon)}{\mathrm{Vol}_q(n, d-1)}.$$

□

Theorem 61 (Bellini-Guerrini-Sala upper bound)**.** *Let* $2 \leq d \leq n$*, let* t *be a positive integer such that* $t \leq \min(n-d, k)$ *with* $k = \left\lceil \log_q(A_q^*(n,d)) \right\rceil$*, let* $r \in \mathbb{N}$ *be such that* $0 \leq r \leq \min(t, \frac{d}{2})$*, and let* $\rho = \frac{q^t}{\mathrm{Vol}_q(t,r)}$*. Then*

$$A_q^*(n, d) \leq \rho \cdot \left(A_q(n-t, d-2r) - \frac{\mathrm{Vol}_q(n-t, r)}{\mathrm{Vol}_q(n-r, d-2r-1)} + 1 \right).$$

Proof. We consider an $(n, M, d)_q$ systematic-embedding code C such that $M = A_q^*(n, d)$. We number all words in C in any order:

$$C = \{c_i \mid 1 \leq i \leq M\}.$$

We indicate the i-th codeword with $c_i = (c_{i,1}, \ldots, c_{i,n})$. We puncture C in a similar way as in proof of the Zinoviev-Litsyn-Laihonen bound:

(i) we choose any t columns among the k columns of the systematic part of C, $1 \leq j_1, \ldots, j_t \leq n$; since any two codes are equivalent w.r.t. column permutations we can suppose $j_1 = 1, \ldots, j_t = t$. Let us split each codeword c_i in two parts, $c_i = (\tilde{c}_i, \bar{c}_i)$, where $\tilde{c}_i$ correspond to the first t components, $\bar{c}_i$ to the remaining $n - t$.

(ii) We choose a $z \in (\mathbb{F}_q)^t$.

(iii) We collect in I all i's such that $\mathrm{d}(z, \tilde{c}_i) \leq r$.

(iv) We delete the first t components of $\{c_i \mid i \in I\}$.

In this way the obtained $\bar{C}_z$ consists of the puncturing of the codewords whose distance in the first t components from a chosen z is at most r:

$$\bar{C}_z = \{\bar{c}_i \mid 1 \leq i \leq A_q^*(n,d),\ \mathrm{d}(z,\tilde{c}_i) \leq r\}.$$

We claim that we can choose z in such a way that $\tilde{C}_z$ is equivalent to a code with the following properties:

1. $\bar{n} = n - t$
2. $\bar{d} \geq d - 2r$
3. $|\bar{C}_z| \geq \frac{M}{q^t}\mathrm{Vol}_q(t,r)$
4. $\mathrm{w}(\bar{c}_i) \geq d - r$ for all $\bar{c}_i \neq 0$.

Properties 1. – 3. comes from the proof of the Zinoviev-Litsyn-Laihonen bound. We claim that 4. holds if $0 \in C$ and $z = 0$. In fact $\mathrm{w}(c) = \mathrm{d}(0,c) \geq d$ for each nonzero codeword c, and $z = 0$ implies that

$$y \in S_r(z) \quad \Longleftrightarrow \quad \mathrm{w}(y) \leq r.$$

As a consequence, any nonzero codeword $c_i = (\tilde{c}_i, \bar{c}_i)$ of weight at most r in $\tilde{c}_i$ has weight at least $d - r$ in the other $n - t$ components. If $0 \notin C$ or $z \neq 0$, we can translate C and we obtain an equivalent code in which $0 \in C$ and $z = 0$, and 4. follows.
We denote with X the largest $(\bar{n}, |X|, d-2r)$ code containing the zero codeword and such that $\mathrm{w}(\bar{x}) \geq d - r = (d - 2r) + r$, $\forall \bar{x} \in X$. Since X satisfies the properties 1. – 4., $|X| \geq |\bar{C}_z|$. Then we can apply Proposition 60 to $X \setminus \{0\}$ and $\varepsilon = r$, and obtain the following chain of inequalities:

$$\frac{|C|}{q^t}\mathrm{Vol}_q(t,r) \leq |\bar{C}_z| \leq |X| \leq A_q(\bar{n}, d-2r) - \frac{\mathrm{Vol}_q(\bar{n},r)}{\mathrm{Vol}_q(\bar{n}, d-2r-1)} + 1,$$

and since $|C| = A_q^*(n,d)$ we conclude the proof. □

Theorem 62 (Bellini-Guerrini-Sala lower bound)**.** *Let $k \geq 1$, $d \geq 2$ and $0 \leq r \leq \frac{d}{2}$. Let n be such that there exists an (n, q^k, d) systematic code. Then*

$$A_q(n-k, d-2r) \geq \mathrm{Vol}_q(k, r) + \frac{\mathrm{Vol}_q(n-k, r)}{\mathrm{Vol}_q(n-k, d-2r-1)} - 1.$$

Proof. If we restrict ourselves to systematic codes, we can replace $A_q^*(n, d)$ in the Bellini-Guerrini-Sala upper bound with q^k for a certain k, and by choosing $t = k$ we conclude the proof. □

Chapter 3

Hadamard Matrices and Codes

> *However, to find these, it has been necessary to construct the very numerous possible combinations, among which the useful ones are to be found.*
>
> Jacques Hadamard

In this chapter we briefly overview an important class of nonlinear codes. In the previous chapter we listed some known bounds on the size of codes, however these bounds are not always attained. In Chapter 5 we will show how the Griesmer bound can be applied to some nonlinear codes (Theorem 96, Corollary 98, Proposition 101 and Theorem 107), and this will lead to new bounds outperforming known bounds (e.g. see Proposition 112). On the other hand the Plotkin bound is known to be attained in most of the cases, and in this section we discuss a class of these optimal codes.
To build these codes we will introduce Hadamard matrices in Section 3.1 and we will define Hadamard codes in Section 3.2. Most of the definitions and results presented in this chapter can be found in [31], [33, Ch. 2,§3] and [36].

3.1 Hadamard matrices

Definition 63. A *Hadamard matrix* of order n is a real $n \times n$ matrix whose entries are either 1 or -1 such that $H \cdot H^t = H^t \cdot H = n \cdot I_n$, where I_n is the identity of order n.

From the definition it follows that both H and H^t are Hadamard matrices and they share the same properties. A first and fundamental property of a Hadamard matrix is that its rows form an orthogonal basis of the space.

Proposition 64. *Let H be a Hadamard matrix. Then the inner product between any two rows is zero, while the inner product between any row with itself is n.*

Two Hadamard matrices are said to be equivalent if one is obtained from the other by swapping rows, transpositions and multiplications of rows and columns by -1. Using the latter, Hadamard matrices can be brought to a so-called *normal form*.

Definition 65. A Hadamard matrix is in *normal form* if the coordinates of its first row and column are all equal to 1.

Theorem 66. *Let H be a Hadamard matrix of order n. Then n is* 1, 2 *or a multiple of* 4.

Proof. The matrices

$$H_1 = \begin{bmatrix} 1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

are respectively Hadamard matrices of order 1 and 2. For the case $n > 2$ we assume without loss of generality that the first three rows of H are in the form

$$\underbrace{\begin{matrix} + + \cdots + \\ + + \cdots + \\ + + \cdots + \end{matrix}}_{i_1} \quad \underbrace{\begin{matrix} + + \cdots + \\ + + \cdots + \\ - - \cdots - \end{matrix}}_{i_2} \quad \underbrace{\begin{matrix} + + \cdots + \\ - - \cdots - \\ + + \cdots + \end{matrix}}_{i_3} \quad \underbrace{\begin{matrix} + + \cdots + \\ - - \cdots - \\ - - \cdots - \end{matrix}}_{i_4}.$$

Since these three rows are orthogonal we obtain the system

$$\begin{cases} i_1 + i_2 - i_3 - i_4 = 0 \\ i_1 - i_2 + i_3 - i_4 = 0 \\ i_1 - i_2 - i_3 + i_4 = 0 \end{cases},$$

which lead us to $i_1 = i_2 = i_3 = i_4$, hence $n = 4i_1$ is a multiple of 4. □

It was conjectured that Hadamard matrices of order n exist whenever n is a multiple of 4. This conjecture, known as *Hadamard Conjecture*, has not been proven yet, even though several constructions for Hadamard matrices are known.

Theorem 67 (Sylvester construction)**.** *Let H_n be a Hadamard matrix. Then*

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \tag{3.1}$$

is a Hadamard matrix of order $2n$.

Proof. We consider two rows h_i and h_j of H_{2n}. We denote them with $h_i = (h'_i|h''_i)$ and $h_j = (h'_j|h''_j)$, where h' is the vector containing the first n coordinates of the row h. We have

$$h_i \cdot h_j = h'_i \cdot h'_j \ + \ h''_i \cdot h''_j,$$

and we have 4 cases:

1. $i = j$ implies that the product is

$$h_i \cdot h_i = h'_i \cdot h'_i \ + \ h''_i \cdot h''_i = n + n = 2n;$$

2. $i = j + n \mod 2n$, which lead to the product

$$h'_i \cdot h'_i \ - \ h''_i \cdot h''_i = n - n = 0;$$

3. $i < j \leq n$ or $n < i < j$. We have

$$h_i \cdot h_j = \begin{cases} h'_i \cdot h'_j \ + \ h''_i \cdot h''_j = 0 + 0, & \text{if } i < j \leq n \\ h'_i \cdot h'_j \ + \ (-h''_i) \cdot (-h''_j) = 0 + 0, & \text{if } n < i < j \end{cases}.$$

4. In the remaining case we have $i \leq n < j$. The vectors h'_i and h'_j are distinct rows of H_n, hence $h'_i \cdot h'_j = 0$. Regarding h''_i and h''_j, we observe that $-h''_j$ is a row of H_n, so $h''_i \cdot (-h''_j) = 0$ which implies $h''_i \cdot h''_j = 0$.

□

Using equation (3.1) iteratively starting from $H_1 = \begin{bmatrix} 1 \end{bmatrix}$ produces Hadamard matrices for all orders equal to powers of 2. These particular matrices are also known as *Sylvester matrices*.

Before presenting the Paley construction we need the following definitions.

Definition 68. Let $\mathbb{F}_q = \{0, \alpha_1, \ldots, \alpha_{q-1}\}$ be a field with odd characteristic. The elements $\alpha_1^2, \ldots, \alpha_{q-1}^2$ are called the *quadratic residues*

of $\mathbb{F}_q$.
The function $\chi : \mathbb{F}_q \to \mathbb{Z}$ defined by

$$\chi(\alpha) = \begin{cases} 0, & \text{if } \alpha = 0 \\ +1, & \text{if } \alpha \text{ is a quadratic residue} \\ -1, & \text{otherwise} \end{cases} \tag{3.2}$$

is called *Legendre symbol*.

Definition 69. Let $\mathbb{F}_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ be a field of odd characteristic. We define $Q = (q_{i,j})$ as the matrix whose entries are $\chi(\alpha_i - \alpha_j)$, where χ is the Legendre symbol. The matrix Q is called *Jacobsthal matrix* of order q.

Theorem 70 (Paley construction)**.** *Let $q = p^m$ be a prime power such that $q = 3 \mod 4$, let $\mathbf{1} = (1, \ldots, 1)$ be the vector of length q whose coordinates are all equal to 1 and let Q be a Jacobsthal matrix of order q. $H = \begin{bmatrix} 1 & \mathbf{1} \\ \mathbf{1}^t & Q - I_q \end{bmatrix}$ is a Hadamard matrix of order $q + 1$*

The proof of Theorem 70 can be found in [33, Ch. 2,§3] for the case of prime fields, and can be directly generalised to the case of non-prime fields. We remark that there exists also another construction due to Paley, which is a modification of Theorem 70 and produces matrices of order $2q + 2$ for each $q = 1 \mod 4$.

3.2 Hadamard codes

Hadamard matrices can be used to obtain optimal binary codes. Given a Hadamard matrix H_n we obtain a binary matrix H_n^b by using the replacement $1 \to 0$ and $-1 \to 1$. H_n^b is known as *binary Hadamard matrix*. If H_n is a normalised matrix, then H_n^b can be seen as a codebook whose first row is the zero codeword and the first coordinate of each codeword is equal to 0. There exist three types of Hadamard codes.

Theorem 71. *Let H_n be a normalised Hadamard matrix and let H_n^b the corresponding binary matrix. The code $\mathcal{A}_n$ obtained by puncturing H_n^b in the first coordinate is an $(n-1, n, \frac{n}{2})_2$ code.*

Proof. Since any two rows of H_n are orthogonal, any two rows of H_n^b differ in exactly $\frac{n}{2}$ coordinates. □

Theorem 72. *Let $1 + \mathcal{A}_n$ be the table whose entries are the complement of the entries of $\mathcal{A}_n$. Let $\mathcal{B}_n$ the code whose codebook is $\begin{bmatrix} \mathcal{A}_n \\ 1 + \mathcal{A}_n \end{bmatrix}$. $\mathcal{B}$ is an $(n-1, 2n, \frac{n}{2} - 1)_2$ code.*

Proof. Let $c_1,\ c_2 \in \mathcal{A}_n$. $\mathrm{d}(c_1, 1 + c_2) = \mathrm{d}(1, c_1 + c_2) = (n-1) - \mathrm{d}(c_1, c_2) = \frac{n}{2} - 1$, while the distance between c_1 and $1 + c_1$ is $n - 1$. □

Theorem 73. *The code $\mathcal{C}_n$ whose codebook is $\begin{bmatrix} H_n^b \\ 1 + H_n^b \end{bmatrix}$ is an $(n, 2n, \frac{n}{2})_2$ code.*

Proof. Let c_1 and c_2 be rows of H_n^b. $\mathrm{d}(c_1, 1 + c_2) = \mathrm{d}(1, c_1 + c_2) = n - \mathrm{d}(c_1, c_2) = \frac{n}{2}$, while $\mathrm{d}(c_1, 1 + c_1) = n$. □

The codes $\mathcal{A}_n$, $\mathcal{B}_n$ and $\mathcal{C}_n$ are known as *Hadamard codes*. We introduce now another family of codes, Simplex codes.

Definition 74. Let $\mathcal{S}_k$ be the binary linear code generated by the matrix G_k whose $2^k - 1$ columns are all the non-zero vectors of $(\mathbb{F}_2)^k$. $\mathcal{S}_k$ is an $[2^k - 1, k, 2^{k-1}]_2$ linear code.

There exist several proofs of the fact that $\mathrm{d}(\mathcal{S}_k) = 2^{k-1}$, here we show it by induction on k.
For $k = 1$ we have $G_1 = \begin{bmatrix} 1 \end{bmatrix}$, the only column of G_1 is the only non-zero vector of $(\mathbb{F}_2)^1$ and the parameters of $\mathcal{S}_1$ are trivially $n = k = d = 1$. Let $\mathbf{0}_k$ be the zero vector of $(\mathbb{F}_2)^k$ and $\mathbf{1}_{2^k-1} = (1, \ldots, 1) \in (\mathbb{F}_2)^{2^k-1}$. Starting from the generator matrix G_k of $\mathcal{S}_k$ we construct

the matrix G_{k+1} by the formula

$$G_{k+1} = \begin{bmatrix} G_k & \mathbf{0}_k^t & G_k \\ 0\cdots 0 & 1 & 1\cdots 1 \end{bmatrix}.$$

G_{k+1} has rank $k+1$ since its $2^{k+1}-1$ columns are all the non-zero vectors of $(\mathbb{F}_2)^{k+1}$, hence G_k generate $\mathcal{S}_{k+1}$, which is an $[2^{k+1}-1, k+1]_2$ linear code. Regarding the distance of this code, let us consider the codewords generated by the first k rows of G_{k+1}. By definition, each of these codewords c is in the form $c = (c', 0, c')$, where $c' \in \mathcal{S}_k$, hence the distance between any pair $c_1 = (c'_1, 0, c'_1)$ and $c_2 = (c'_2, 0, c'_2)$ are at distance $2 \cdot \mathrm{d}(c'_1, c'_2) \geq 2 \cdot \mathrm{d}(\mathcal{S}_k) = 2^k$.
It remains to prove that the distance between the last row g_{k+1} of G_{k+1} and any of the codewords $c = (c', 0, c')$ as above are at distance at least 2^k. We know that the weight of c' is $w \geq 2^k$, and $\mathrm{d}(1\cdots 1, c') = 2^k - 1 - w$, hence $\mathrm{d}(g_{k+1}, c) = w + 1 + 2^k - 1 - w = 2^k$.
This concludes the proof of $\mathrm{d}(\mathcal{S}_k) = 2^{k-1}$. Moreover, from this we directly obtain the following theorem.

Theorem 75. *$\mathcal{S}_k$ is an equidistant code, namely each pair c_1, c_2 of codewords are at distance $d = 2^{k-1}$. In particular all non-zero codewords have weight equal to the minimum distance 2^{k-1}.*

Simplex codes and Hadamard codes are deeply connected. If we add the bit 0 at the beginning of each codeword of $\mathcal{S}_k$ we obtain an $[2^k, k, 2^{k-1}]_2$ code. Its codebook is a square $2^k \times 2^k$ table, and each pair of its rows are at distance 2^{k-1}, namely its codebook is a binary Hadamard matrix $H^n_{2^k}$. This also implies that $\mathcal{S}_k$ is a Hadamard code $\mathcal{A}_{2^k}$ obtained using Theorem 71.

We recall now the binary Plotkin bound in Theorem 48. Each binary code with even distance and with $n < 2d$ has size at most

$$A_2(n, d) \leq 2 \left\lfloor \frac{d}{2d - n} \right\rfloor.$$

In the case of $n = 2^k - 1$ and $d = 2^{k-1}$ as for Simplex codes, we have

$$A_2\left(2^k - 1, 2^{k-1}\right) \leq 2\left\lfloor \frac{2^{k-1}}{2 \cdot 2^{k-1} - 2^k + 1} \right\rfloor = 2^k.$$

This shows that Simplex codes are optimal codes. In general we have the following result.

Proposition 76. *Hadamard codes A_n are optimal.*

Proof. It has already been proved above for Simplex codes, which are Hadamard $\mathcal{A}_{2^k}$ codes. For Hadamard codes $\mathcal{A}_n$ we observe that

$$A_2\left(n - 1, \frac{n}{2}\right) \leq 2\left\lfloor \frac{\frac{n}{2}}{2 \cdot \frac{n}{2} - n + 1} \right\rfloor = n.$$

□

Hadamard codes and Simplex codes are therefore of paramount importance since they allow us to understand the parameters and properties of optimal codes, at least in the Plotkin range. In particular the existence of Simplex codes implies the existence of linear codes attaining the Plotkin bound, even though in the general case nonlinear codes could achieve better error correction capability than linear codes, as we will see in Chapter 5.

Proposition 77. *For each pair of positive integers k and h, there exists an $\left[h\left(2^k - 1\right), k, h2^{k-1}\right]_2$ optimal code.*

Proof. Let G_k be the generator matrix of the Simplex code $\mathcal{S}_k$. The claimed code is generated by

$$G = \left[\underbrace{G_k \quad \cdots \quad G_k}_{h \text{ times}}\right] \tag{3.3}$$

□

We observe in particular that these codes attain the Plotkin bound, in fact we have the following lemma.

Lemma 78. *Let n, d and h be positive integers and let $(n, d) = 1$. Let C be an $(n, M, d)_q$ code attaining the Plotkin bound in Theorem 47. The code whose codebook is*

$$\left[\underbrace{C \quad \cdots \quad C}_{h \text{ times}}\right]$$

attains itself the Plotkin bound and is therefore optimal.

Proof. It follows directly from the Plotkin bound. □

Regarding $\left[h\left(2^k - 1\right), k, h2^{k-1}\right]_2$ linear codes, namely codes with parameters equal to that of sequences of Simplex codes, there exists another important result due to Bonisoli [8].

Theorem 79. *Any linear code with length $h\left(2^k - 1\right)$, dimension k and distance $h2^{k-1}$ is equivalent to a code as in Equation (3.3).*

This implies that, up to equivalence, the only optimal linear codes with parameters as in Proposition 77 are Simplex codes.

We conclude this chapter with a remarkable result on optimal codes, proven in [31] by Levenshtein.

Theorem 80. *Provided enough Hadamard matrices exist, the binary Plotkin bound is attained with equality.*

Before providing the proof, we need a few lemmas.

Lemma 81. *Let $\ddot{\mathcal{A}}_n$ be a Hadamard code as in Theorem 71. The code $\ddot{\mathcal{A}}_n$ obtained by shortening $\mathcal{A}_n$ in the first coordinate is an $\left(n-2, \frac{n}{2}, \frac{n}{2}\right)_2$ code.*

Proof. Any column of a Hadamard matrix contains half zeros and half ones. □

Lemma 82. *Let h_1 and h_2 be two positive integers and let C_1, C_2 be respectively an $(n_1, M_1, d_1)_2$ and an $(n_2, M_2, d_2)_2$ code.*

Let $h_1C_1||h_2C_2$ be the code whose codebook is given by

$$\left[\underbrace{C_1 \quad \cdots \quad C_1}_{h_1 \text{ times}} \quad \underbrace{C_2 \quad \cdots \quad C_2}_{h_2 \text{ times}}\right],$$

where if $M_1 \neq M_2$ we consider only a subcode of the code with the larger size, so that $h_1C_1||h_2C_2$ has size $M = \min(M_1, M_2)$.
The obtained code is an $(h_1n_1 + h_2n_2, M, d)_2$ code, where $d \geq h_1d_1 + h_2d_2$.

We can give now the proof of Theorem 80.

Proof. First, we recall that we can restrict ourselves to the case of d even, since the case d odd follows from the the codes of distance $d+1$ by puncturing.
Let $n < 2d$, and let $k = \left\lfloor \frac{d}{2d-n} \right\rfloor$. Let us define

$$h_1 = d\,(2k+1) - n\,(k+1)\,, \quad h_2 = kn - d\,(2k-1)\,.$$

We observe that both h_1 and h_2 are non-negative integers. If n is even so are h_1 and h_2, while if n is odd and if k is odd then h_1 is even, otherwise h_2 is even. Moreover,

$$n = (2k-1)\,h_1 + (2k+1)\,h_2, \quad d = kh_1 + (k+1)\,h_2.$$

If n is even, then

$$C = \frac{h_1}{2}\ddot{\mathcal{A}}_{4k}||\frac{h_2}{2}\ddot{\mathcal{A}}_{4k+4}.$$

If instead n is odd, we have the following two codes depending on the parity of k.
If k is odd, then

$$C = \frac{h_1}{2}\ddot{\mathcal{A}}_{4k}||h_2\mathcal{A}_{2k+2},$$

while if k is even

$$C = h_1\mathcal{A}_{2k}||\frac{h_2}{2}\ddot{\mathcal{A}}_{4k+4}.$$

Either way, C is an optimal code attaining the Plotkin bound with equality. □

Chapter 4

Introduction to Random Number Generation

> *There are a number of difficult epistemological questions connected with the theory of secrecy, or in fact with any theory which involves questions of probability.*
>
> Claude Shannon

Random Numbers are of great importance in many fields of application, among which we list gaming, simulation of physical phenomena for research purposes, and security protocols for safe communications. Devices able to provide these random numbers are called *Random Number Generators* (RNGs) and are classically distinguished between *pseudo-RNGs* and *True-RNGs*. Other classifications take into account whether they are algorithms or physical devices, or whether they make use of chaotic or quantum physical phenomena to generate data. In this work we refer to [2], [3], [4] and [6] to have precise definitions, however we also give the following intuitive definition.

Definition 83. A random bit is the unpredictable result of an experiment with two possible outcomes.
A Random Number Generator is a physical device able to output sequences of random bits.

To be more precise, RNGs are made by several components:

- A *noise source*,
- A *digitisation process*,
- An optional *entropy extractor*.

Definition 84. A noise source is a discrete-time discrete-space stochastic process X_t. In this work we consider only IID noise sources, and we denote with μ_X the probability mass function (pmf) of X_t.
We denote with U_q the uniform distribution on a discrete set with q elements.

Assume $X_t \in \mathbb{F}_2$, namely we are dealing with a *binary* noise source. In this case $\mu_X(1) = p$ and $\mu_X(0) = 1 - p$. If $p = \frac{1}{2}$ then $\mu_X = U_2$ and X_t is *uniformly distributed*, or *unbiased*.
If X_t is not unbiased, we need a way to measure its distance from the uniform distribution.

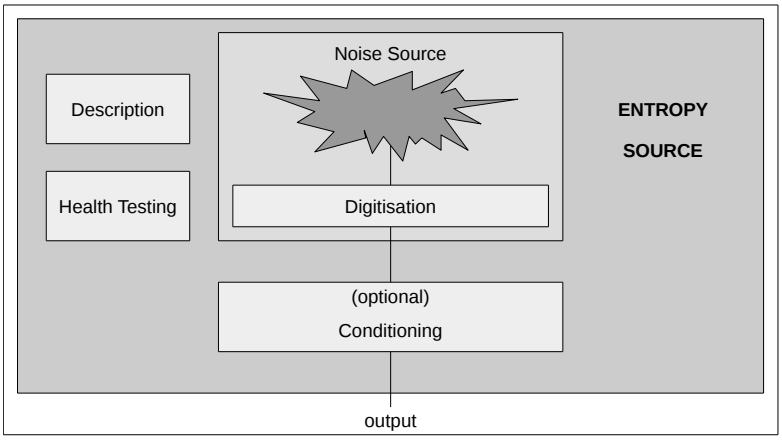

Figure 4.1: Structure of a RNG: Barker, Elaine, and John Kelsey. *NIST DRAFT SP800-90B, Recommendation for the entropy sources used for random bit generation* (2012).

Definition 85. The *bias* of a binary random variable X is defined as

$$\frac{\varepsilon_X}{2} = \frac{1}{2}\left|\mathbb{P}(X=1) - \mathbb{P}(X=0)\right| = \frac{1}{2}\left|\mu_X(1) - \mu_X(0)\right|.$$

Since $\mu_X(0) = 1 - \mu_X(1)$ and $\mathbb{E}(X) = \mu_X(1)$, we also have

$$\frac{\varepsilon_X}{2} = \frac{1}{2}\left|2\mathbb{E}[X] - 1\right|.$$

Assume now w.l.o.g. that $\mu_X(1) > \mu_X(0)$. Then

$$\begin{aligned} \left|\mu_X(1) - \mu_X(0)\right| &= \left|\mu_X(1) - \tfrac{1}{2} + \tfrac{1}{2} - \mu_X(0)\right| \\ &= \left|\mu_X(1) - \tfrac{1}{2}\right| + \left|\tfrac{1}{2} - \mu_X(0)\right| \\ &= \left\|\mu_X - U_2\right\|_1 \end{aligned}$$

Definition 86. The *Total Variation distance* (TVD) of a random variable $X \in \Omega_X$, with $|\Omega_X| = q$, is defined as

$$\frac{\delta_X}{2} = \frac{1}{2}\left\|\mu_X - U_q\right\|_1.$$

From this definition we have that the bias is a particular case of the TVD. Another measure of the distance from the uniform distribution is the *entropy* of a random variable.

Definition 87. Given $X \in \Omega_X$ and its probability mass function μ_X, the *entropy* of X is defined as

$$H(X) = -\mathbb{E}[\log_q(\mu_X)]$$

Proposition 88.

- $H(X) \in [0, 1]$.
- $H(X) = 0$ *if and only if* X *is deterministic.*
- $H(X) = 1$ *if and only if* X *is uniformly distributed,* $\mu_X = U_q$.

These imply that $H(X) = 1$ *if and only if* $\frac{\delta_X}{2} = 0$.

Consider again a noise source X_t. If $\frac{\delta_X}{2} \neq 0$ then X_t is not uniformly distributed. In this case the output of X_t should not be used for security-related applications: suppose we are using X_t to produce keys for a symmetric cipher. Since X_t is not uniformly distributed so are the generated keys, hence an attacker could exploit this weakness to obtain informations on the communication. To solve this problem, raw sequences produced by the noise source X_t have to be *processed*, in order to obtain an output whose probability distribution is (as close as possible to) U_q. Deterministic functions designed to address this task are known as *conditioning functions*, *post-processing* or *entropy extractors*.

4.1 Entropy extractors

Definition 89. An *entropy extractor* is a function $\varphi : \Omega_X \to \Omega_Y$. Given a random variable $X \in \Omega_X$, we denote with $Y = \varphi(X)$ the output of φ.

The aim of an entropy extractor is to output a random variable Y whose entropy is greater than the entropy of X.

Proposition 90. *if* $|\Omega_Y| \geq |\Omega_X|$, *then* $H(Y) \leq H(X)$.

Due to Proposition 90, φ has to be a *compression function*, i.e. $|\Omega_Y| < |\Omega_X|$.

4.1.1 Von Neumann procedure

The Von Neumann procedure is an entropy extractor for binary random number generators. Suppose an IID RNG produces binary output. Then given a pair (x_t, x_{t+1}), it holds

$$\mathbb{P}((x_t, x_{t+1}) = (0, 1)) = \mathbb{P}((x_t, x_{t+1}) = (1, 0)).$$

Von Neumann extractor groups together non-overlapping pairs of consecutive bits, then for each pair it outputs a single bit, according to the rule:

- If the two bits in the pair have different values, output the first bit.
- If the two bits are equal, discard the pair

Further works have extended the idea of this extractor to reiterated procedures, in which the same extractor is applied to the sequences of discarded pairs.
Algorithm 1 is a pseudo-code for the Von Neumann procedure.

4.1.2 Binary linear extractors

Binary linear extractors are a well known class of extractors deeply investigated in literature in case of an entropy source assumed to produce biased independent bits (see e.g. [28] and [29]) and accepted by NIST in the revised version of [2], which is under review and will substitute the current version of the recommendation.
Let X be a binary noise source able to produce independent bits and let G be a $k \times n$ binary matrix. We remark that in this section, as we will do in Chapter 6, vectors are considered as column vectors to simplify the notation.

ALGORITHM 1:
Data: $X_T \in \mathbb{F}_2$, $T \in [1, N_0]$
Result: Y_s, Z_q
Initialization:
$N_1 = \lfloor \frac{N_0}{2} \rfloor$;
Build the process $(X_{2t-1}, X_{2t})_{t\in[1,N_1]}$;
$t = 1$;
$s = 1$;
$q = 1$;
while $t \leq N_1$ **do**
  if $(X_{2t-1} == X_{2t})$ **then**
    $Z_q = X_{2t-1}$;
    $q = q + 1$;
  else
    if $X_{2t-1} == 0$ **then**
      $Y_s = 0$;
    else
      $Y_s = 1$;
    end
    s=s+1
  end
  $t = t + 1$;
end

Algorithm 1: A pseudo-code for the Von Neumann procedure. The output of this code are the extracted sequence Y_s and a sequence Z_q which contains a representative of each discarded bit pair $X_{2t-1} == X_{2t}$. By using this algorithm itself on Z_q we can obtain the reiterated version of the procedure.

Definition 91. Let $\bar{X}$ be a binary random vector of length n obtained by collecting random bits from the noise source X. A binary linear extractor is a map $(\mathbb{F}_2)^n \to (\mathbb{F}_2)^k$ defined by

$$\bar{X} \mapsto \bar{Y} = G \cdot \bar{X}.$$

The resulting binary vector $\bar{Y} = (Y_1, \ldots, Y_n)^T$ is not a vector whose coordinates are independent random bits, however it is a random binary vector whose probability mass function depends on both the bias of X and the properties of the matrix G. To be effective, this extractor has to be a compression function, hence $k < n$.
Complete proofs of the following theorems can be found in [28].

Theorem 92 (Lacharme)**.** *The bias of any non-zero linear combination of $Y_1, \ldots, Y_k$ is bounded by $\frac{\varepsilon_X^d}{2}$, where*

- *ε_X is the bias of X_i*
- *d is the minimum distance of the code generated by G.*

Proof. We provide here a sketch of the proof.
Any non-zero linear combination of $Y_1, \ldots, Y_k$ can be seen as the sum of at least d elements among $X_1, \ldots, X_n$. Let $\mathbb{P}(X_i = 1) = p = \frac{1+\varepsilon_X}{2}$. The probability of the sum of two terms is given by

$$\begin{aligned} \mathbb{P}(X_{i_1} + X_{i_2} = 1) &= \mathbb{P}(X_{i_1} \neq X_{i_2}) \\ &= 2p(1-p) \\ &= 2\left(\tfrac{1+\varepsilon_X}{2}\right)\left(\tfrac{1-\varepsilon_X}{2}\right) \\ &= \tfrac{1-\varepsilon_X^2}{2} \end{aligned}$$

The bias of the sum is given by

$$\begin{aligned} \tfrac{1}{2}\left|2\mathbb{P}(X_{i_1} + X_{i_2} = 1) - 1\right| &= \tfrac{1}{2}\left|2\tfrac{1-\varepsilon_X^2}{2} - 1\right| \\ &= \tfrac{\varepsilon_X^2}{2}, \end{aligned}$$

hence we can conclude by induction. □

We conclude this section with another result on linear extractors due to Lacharme. The proof of the following theorem can be found in [28]. Let $\frac{\delta_{\bar{Y}}}{2}$ be the TVD from the uniform distribution of $\bar{Y} = G\bar{X}$.

Theorem 93 (Lacharme, 2008)**.**

$$\delta_{\bar{Y}} \leq (2^k - 1)\varepsilon_X^d.$$

These two results show an interesting connection between binary linear extractors and linear codes. We will investigate more deeply this fact in Chapter 6.2.

Part II

Main Results

Chapter 5

On optimal systematic codes

> *All results of the profoundest mathematical investigation must ultimately be expressible in the simple form of properties of the integers.*
>
> Leopold Kronecker

In Section 2.5 we introduced the Griesmer bound, a well known result on linear codes bounding the length of optimal codes. We will often refer to it with the notation $g_q(k,d)$, namely

$$g_q(k,d) := \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil .$$

Many papers, such as [18], [20], [27], [35], [46], and have characterized classes of linear codes meeting the Griesmer bound. In particular, finite projective geometries play an important role in the study of these codes. For example in [17], [21] and [47] minihypers and maxhypers are used to characterize linear codes meeting the Griesmer bound. Research has been done also to characterize the codewords of linear codes meeting the Griesmer bound [51].
Many known bounds on the size of codes, for example the Johnson bound [23], [24], [25], the Elias-Bassalygo bound [5], [23], the Hamming (Sphere Packing) bound, the Singleton bound [38], the Zinoviev-Litsyn-Laihonen bound [30], [53], the Bellini-Guerrini-Sala bound [7], and the Linear Programming bound [11], are true for both linear and (systematic) nonlinear codes.
On the other hand, the proof of the Griesmer bound heavily relies on the linearity of the code and it cannot be applied to all codes.

Most of the results presented here can also be found in [16]. In this chapter we present our results on systematic codes and their relations to (possible extensions of) the Griesmer bound. We will start by proving that, once q and d have been chosen, if all nonlinear $(n, q^k, d)_q$ systematic codes with $k < 1+\log_q d$ respect the Griesmer bound, then the Griesmer bound holds for all systematic codes with the same q and d. Therefore, for any q and d only a finite set of (k,n) pairs has to be analysed in order to prove the bound for all k and n. In Section 5.1 we identify several families of parameters for which the Griesmer bound holds in the systematic (nonlinear) case. In Section 5.2 we

provide some versions of the Griesmer bound holding for systematic codes.
In the next sections we study optimal binary codes with small size, namely $M = 4$ and $M = 8$. In Section 5.3 we show that all optimal binary codes with 4 codewords are necessarily (equivalent to) linear codes. In Section 5.4 we show that for any possible distance, there exist binary linear codes with 8 codewords achieving the Plotkin bound, and this implies that $N_2(8, d) = S_2(3, d) = L_2(3, d)$. Finally, in Section 5.5, we show explicit counterexamples of binary systematic codes for which the Griesmer bound does not hold, by constructing a family of optimal binary systematic codes. In the final section we draw our conclusions and hint at a future work and open problems.

In this work we consider the following definition of an optimal code.

Definition 94. Let k and d be two positive integers. An $(n, M, d)_q$ code C is *optimal* if all codes with the same distance and size have length at least n.
An $(n, q^k, d)_q$ systematic code C is *optimal* if all systematic codes with the same distance and dimension have length at least n.

Theorem 95. *For fixed q and d, if*

$$S_q(k, d) \geq g_q(k, d) \tag{5.1}$$

for all k such that $1 \leq k < 1 + \log_q d$, then (5.1) holds for any k, i.e. the Griesmer bound is true for all systematic codes over $\mathbb{F}_q$ with minimum distance d.

Before proving it, we remark that an equivalent formulation for Theorem 95 could be: *If there exists an $(n, q^k, d)_q$ systematic code which does not satisfy the Griesmer bound, then there exists an $(n', q^{k'}, d)_q$ systematic code with $k' < 1 + \log_q d$ which does not satisfy the Griesmer bound.*

Proof. For each fixed d and q, suppose there exists an $(n, q^k, d)_q$ systematic code not satisfying the Griesmer bound, i.e., there exists k such that $S_q(k,d) < g_q(k,d)$. Let us call $\Lambda_{q,d} = \{k \geq 1 \mid S_q(k,d) < g_q(k,d)\}$.
If $\Lambda_{q,d}$ is empty then the Griesmer bound is true for such parameters q, d.
Otherwise, there exists a minimum $k' \in \Lambda_{q,d}$ such that $S_q(k',d) < g_q(k',d)$.
In this case we can consider an $(n, q^{k'}, d)_q$ systematic code C not verifying the Griesmer bound, $n = S_q(k',d)$.
We obtain an $(n-1, q^{k'-1}, d')$ systematic code C' whose distance is $d' \geq d$ by applying Proposition 23 to C, then we apply Lemma 24 to C', hence we obtain an $(n-1, q^{k'-1}, d)_q$ systematic code $\bar{C}$.
Since k' was the minimum among all the values in $\Lambda_{q,d}$, then the Griesmer bound holds for $\bar{C}$, and so

$$n - 1 \geq g_q(k'-1, d) = \sum_{i=0}^{k'-2} \left\lceil \frac{d}{q^i} \right\rceil . \tag{5.2}$$

We observe that, if $q^{k'-1} \geq d$, then $\left\lceil \frac{d}{q^{k'-1}} \right\rceil = 1$, so we can rewrite (5.2) as

$$n \geq \sum_{i=0}^{k'-2} \left\lceil \frac{d}{q^i} \right\rceil + 1 \geq \sum_{i=0}^{k'-2} \left\lceil \frac{d}{q^i} \right\rceil + \left\lceil \frac{d}{q^{k'-1}} \right\rceil = \sum_{i=0}^{k'-1} \left\lceil \frac{d}{q^i} \right\rceil = g_q(k', d)$$

Since we supposed $n < g_q(k',d)$, we have reached a contradiction with the assumption $q^{k'-1} \geq d$. Hence for such d, the minimum k in $\Lambda_{q,d}$ must satisfy $q^{k-1} < d$, which is equivalent to our claimed expression $k < 1 + \log_q d$. □

5.1 The Griesmer bound and systematic codes

In this section we identify several sets of parameters (q,d) for which the Griesmer bound holds for systematic codes. Subsections

5.1.1 and 5.1.2 deal with q-ary codes, while in Subsection 5.1.3 we consider the special case of binary codes.

5.1.1 The case $d \leq 2q$

Theorem 96. *If $d \leq 2q$ then $S_q(k,d) \geq g_q(k,d)$.*

Proof. First, consider the case $d \leq q$. By Theorem 95 it is sufficient to show that, fixing q and d, for any n there is no $(n, q^k, d)_q$ systematic code with $1 \leq k < 1 + \log_q d$ and $n < g_q(k,d)$. If $1 \leq k < 1 + \log_q d$ then $\log_q d \leq \log_q q = 1$, and so k may only be 1. Since $g_q(1,d) = d$ and $n \geq d$, we clearly have that $n \geq g_q(1,d)$.

Now consider the case $q < d \leq 2q$. If $1 \leq k < 1 + \log_q d$ then $\log_q d \leq \log_q 2q = 1 + \log_q 2$, and so k can only be 1 or 2. We have already seen that if $k = 1$ then $n \geq g_q(k,d)$ for any n, so suppose $k = 2$. If an $(n, q^2, d)_q$ systematic code C exists with $n < \sum_{i=0}^{1} \left\lceil \frac{d}{q^i} \right\rceil = d + 2$, then by the Singleton bound we can only have $n = d + 1$. Therefore C must have parameters $(d+1, q^2, d)$. In [22, Ch. 10] it is proved that a q-ary $(n, q^2, n-1)_q$ code is equivalent to a set of $n-2$ mutually orthogonal Latin squares (MOLS) of order q, and that there are at most $q-1$ Latin squares in any set of MOLS of order q [22, Theorem 10.18]. In our case $n = d+1 > q+1$, therefore $n-2 > q-1$. The existence of C would imply the existence of a set of more than $q-1$ MOLS, which is impossible. □

5.1.2 The case $q^{k-1} \mid d$

The following proposition is a simple consequence of the Plotkin bound that implies some results on values for the distance and dimension for which the Griesmer bound holds in the nonlinear case. We will also make use of this result to obtain a version of the Griesmer bound which can be applied to all systematic codes.

Proposition 97. *If $q^{k-1} \mid d$, then the Griesmer bound coincides with the Plotkin bound.*

Proof. If $q^{k-1} \mid d$, then $g_q(k,d) = \sum_{i=0}^{k-1} \frac{d}{q^i} = d\sum_{i=0}^{k-1} \frac{1}{q^i} = d\frac{1-\frac{1}{q^k}}{1-\frac{1}{q}}$. □

Corollary 98. *Let $r \geq 1$, then $N_q(q^k, q^{k-1}r) \geq g_q(k, q^{k-1}r)$.*

Proof. Follows directly from Proposition 97. □

Note that Corollary 98 is not restricted to systematic codes, and holds for any code with at least q^k codewords, so we can obtain directly the next corollary.

Corollary 99. *Let $M \geq q^k$ and $r \geq 1$, then*

$$N_q(M, q^{k-1}r) \geq g_q(k, q^{k-1}r).$$

The following lemma holds for any nonlinear code.

Lemma 100. *Let $1 \leq r < q$, $l \geq 0$, $d = q^l r$ and let $q^{k-1} \leq d$. Then $N_q(q^k, d) \geq g_q(k, d)$.*

Proof. Since $1 \leq r < q$, the hypothesis $q^{k-1} \leq d$ is equivalent to $k - 1 \leq l$, hence $q^{k-1} \mid d$ and we can apply Proposition 97. □

Proposition 101. *Let $1 \leq r < q$ and $l \geq 0$. Then $S_q(k, q^l r) \geq g_q(k, q^l r)$.*

Proof. Due to Theorem 95 we only need to prove that the Griesmer bound is true for all choices of k such that $q^{k-1} \leq d$. Then we can use Lemma 100, which ensures that all such codes respect the Griesmer bound. □

Corollary 102. *Let $q = 2$ and $l \geq 0$. Then $S_2(k, 2^l) \geq g_2(k, 2^l)$.*

Proof. It follows directly from Proposition 101, with $r = 1$. □

5.1.3 The case $q = 2$, $d = 2^r - 2^s$

In this section we prove that the Griesmer bound holds for all binary systematic codes whose distance is the difference of two powers of 2. We need the following lemmas.

Lemma 103. *Let $r \geq 0$ and let $k \leq r + 1$. Then*

$$g_2(k, 2^{r+1}) = 2g_2(k, 2^r).$$

Proof. The hypothesis $k \leq r + 1$ implies that for any $i \leq k - 1$, both $\left\lceil \frac{2^{r+1}}{2^i} \right\rceil = \frac{2^{r+1}}{2^i}$ and $\left\lceil \frac{2^r}{2^i} \right\rceil = \frac{2^r}{2^i}$. Therefore

$$g_2(k, 2^{r+1}) = \sum_{i=0}^{k-1} \left\lceil \frac{2^{r+1}}{2^i} \right\rceil = \sum_{i=0}^{k-1} \frac{2^{r+1}}{2^i} = 2\sum_{i=0}^{k-1} \frac{2^r}{2^i} = 2\sum_{i=0}^{k-1} \left\lceil \frac{2^r}{2^i} \right\rceil = 2g_2(k, 2^r)$$

□

Lemma 104. *Let $l \geq 0$ be the maximum integer such that 2^l divides d. Then*

$$g_2(k, d + 1) = g_2(k, d) + \min(k, l + 1), \tag{5.3}$$

Proof. Clearly $d = 2^l r$, where r is odd, and the Griesmer bound becomes

$$g_2(k, d + 1) = \sum_{i=0}^{k-1} \left\lceil \frac{2^l r + 1}{2^i} \right\rceil. \tag{5.4}$$

We consider first the case $k \leq l + 1$, and we observe that for each i we have

$$\left\lceil \frac{2^l r + 1}{2^i} \right\rceil = \frac{2^l r}{2^i} + \left\lceil \frac{1}{2^i} \right\rceil = \frac{2^l r}{2^i} + 1 = \left\lceil \frac{2^l r}{2^i} \right\rceil + 1.$$

Therefore

$$g_2(k, d + 1) = \sum_{i=0}^{k-1} \left(\left\lceil \frac{2^l r}{2^i} \right\rceil + 1 \right) = g_2(k, d) + k. \tag{5.5}$$

If $k > l+1$ we can split the sum (5.4) in the two following sums:

$$g_2(k, d+1) = \left(\sum_{i=0}^{l} \left\lceil \frac{2^l r + 1}{2^i} \right\rceil \right) + \left(\sum_{i=l+1}^{k-1} \left\lceil \frac{2^l r + 1}{2^i} \right\rceil \right). \tag{5.6}$$

For the first sum we make use of the same argument as above, while for the second sum we observe that $i > l$, which implies

$$\left\lceil \frac{2^l r + 1}{2^i} \right\rceil = \left\lceil \frac{2^l r}{2^i} \right\rceil.$$

Putting together the two sums, equation (5.6) becomes

$$\begin{aligned} g_2(k, d+1) &= \left(\textstyle\sum_{i=0}^{l} \left\lceil \frac{2^l r}{2^i} \right\rceil + l + 1 \right) + \left(\textstyle\sum_{i=l+1}^{k-1} \left\lceil \frac{2^l r}{2^i} \right\rceil \right) \\ &= \textstyle\sum_{i=0}^{k-1} \left\lceil \frac{2^l r}{2^i} \right\rceil + l + 1, \end{aligned}$$

and the term on the right-hand side is $g_2(k, d) + l + 1$. Together with (5.5) this concludes the proof. □

Lemma 105. *Let k, r and s be integers such that $r > s$ and $k > s+1$. Then*

$$g_2(k, 2^r) - g_2(k, 2^r - 2^s) = 2^{s+1} - 1.$$

Proof. For any d' in the range $2^r - 2^s \leq d' < 2^r$ we can apply Lemma 104, observing that $d' = 2^l \rho$ where $\rho \nmid d'$ and $l \leq s$, which implies $k > l+1$. In particular we observe that $d' = 2^r - \delta$ for a certain $\delta \leq 2^s$, and since 2^l has to divide both 2^r and δ it follows that l depends only on the latter. For a fixed δ we denote with l_δ the corresponding exponent.

From Lemma 104 we obtain

$$g_2(k, 2^r - \delta + 1) = g_2(k, 2^r - \delta) + l_\delta + 1.$$

Applying it for all distances from $2^r - 2^s$ to 2^r we obtain

$$g_2(k, 2^r) - g_2(k, 2^r - 2^s) = \sum_{\delta=1}^{2^s} (l_\delta + 1) = \sum_{\delta=1}^{2^s} l_\delta + 2^s. \tag{5.7}$$

For each value of s, we call $L_s = (l_1, \ldots, l_{2^s})$ the sequence of integers $\{l_\delta\}$ that appear in equation (5.7), and with T_s the sum itself, so that we can write equation (5.7) as

$$g_2(k, 2^r) - g_2(k, 2^r - 2^s) = T_s + 2^s.$$

In the following we will prove that $T_s = 2^s - 1$. First, we show that $L_s = (l_1, \ldots, l_{2^s})$ is equal to

$$(l_1, \ldots, l_{2^{s-1}}, l_1, \ldots, l_{2^{s-1}-1}, l_{2^{s-1}} + 1),$$

namely the first 2^{s-1} terms are exactly the sequence L_{s-1}, while the second half of the sequence is itself equal to L_{s-1} with the exception of the last term, which is incremented by 1.
The fact that the first 2^{s-1} elements of L_s are the elements of L_{s-1} follows directly from the definition of L_s, since l_δ is the largest integer such that $2^{l_\delta} \mid \delta$. For the same reason, $l_{2^s} = l_{2^{s-1}} + 1$. We take now an element in the second half of L_s, which can be written as $l_{2^{s-1}+\bar{\delta}}$, for a certain $1 \leq \bar{\delta} \leq 2^{s-1}$. Using the same argument as before, the integer $l_{2^{s-1}+\bar{\delta}}$ depends only on $\bar{\delta}$ and is equal to $l_{\bar{\delta}}$. Some examples of L_s are listed in Table 5.1.

s	L_s
1	(0,1)
2	(0,1,0,2)
3	(0,1,0,2,0,1,0,3)
4	(0,1,0,2,0,1,0,3,0,1,0,2,0,1,0,4)

Table 5.1: The sequences L_s for $s = 1, 2, 3, 4$.

From the properties of L_s it follows that $T_s = 2T_{s-1} + 1$. Using induction on s, with first step $T_1 = 2^1 - 1$, we now prove our claim $T_s = 2^s - 1$: if $T_{s-1} = 2^{s-1} - 1$, then

$$T_s = 2T_{s-1} + 1 = 2\left(2^{s-1} - 1\right) + 1 = 2^s - 1. \tag{5.8}$$

Putting together equations (5.7) and (5.8) we obtain

$$g_2(k, 2^r) - g_2(k, 2^r - 2^s) = 2^s - 1 + 2^s = 2^{s+1} - 1.$$

□

Lemma 106. *If $k \leq r$, then $g_2(k, 2^r) < 2^{r+1}$.*

Proof. Due to $k \leq r$, for $i < k$ it holds $\left\lceil \frac{2^r}{2^i} \right\rceil = \frac{2^r}{2^i}$. We can write the Griesmer bound as

$$g_2(k, 2^r) = \sum_{i=0}^{k-1} \frac{2^r}{2^i} = 2^r \sum_{i=0}^{k-1} \frac{1}{2^i} < 2^r \cdot 2.$$

□

Theorem 107. *Let r and s be integers such that $r > s \geq 1$ and let $d = 2^r - 2^s$. Then $S_2(k, d) \geq g_2(k, d)$.*

Proof. If $r = s + 1$, then $2^r - 2^s = 2^s$, hence we can apply Corollary 102 and our claim holds. Therefore we can assume $r \geq s + 2$ in the rest of the proof.

Our proof is by contradiction, by supposing that $S_2(k, 2^r - 2^s) < g_2(k, 2^r - 2^s)$, i.e. the Griesmer bound does not hold for some $(n, 2^k, d)_2$ systematic code C, with $d = 2^r - 2^s$ and $n = S_2(k, d)$. Due to Theorem 95, we can assume that $k < 1 + \log_2 d$ and so $k \leq r$.
We call m the ratio n/d, which in the case of C is

$$m = \frac{S_2(k, 2^r - 2^s)}{2^r - 2^s} \leq \frac{g_2(k, 2^r - 2^s) - 1}{2^r - 2^s} \tag{5.9}$$

We claim that

$$m < \frac{g_2(k, 2^r)}{2^r}. \tag{5.10}$$

First we observe that since $k \leq r$, then

$$\frac{g_2(k, 2^r)}{2^r} = \sum_{i=0}^{k-1} \frac{1}{2^i} = 2\left(1 - \frac{1}{2^k}\right).$$

We consider now the ratio m:

$$m \leq \frac{g_2(k, 2^r - 2^s) - 1}{2^r - 2^s} = \frac{1}{2^r - 2^s} \sum_{i=0}^{k-1} \left\lceil \frac{2^r - 2^s}{2^i} \right\rceil - \frac{1}{2^r - 2^s} \quad (5.11)$$

We consider first the case $k \leq s + 1$, and we can write (5.11) as

$$m < \frac{1}{2^r - 2^s} \sum_{i=0}^{k-1} \frac{2^r - 2^s}{2^i} = \sum_{i=0}^{k-1} \frac{1}{2^i} = 2\left(1 - \frac{1}{2^k}\right),$$

so in this case $m < \frac{g_2(k,2^r)}{2^r}$, which is exactly claim (5.10).

We consider now the case $k \geq s + 2$. To prove (5.10), we prove that the term on the right-hand side of inequality (5.9) is itself less than $\frac{g_2(k,2^r)}{2^r}$, and we write this claim in the following equivalent way:

$$2^r(g_2(k, 2^r - 2^s) - 1) < (2^r - 2^s)g_2(k, 2^r).$$

Rearranging the terms we obtain

$$2^s g_2(k, 2^r) < 2^r(g_2(k, 2^r) - g_2(k, 2^r - 2^s) + 1) = 2^r \cdot 2^{s+1}, \quad (5.12)$$

where the equality on the right hand side is obtained from Lemma 105. Hence

$$g_2(k, 2^r) < 2^{r+1},$$

and this is always true provided $k \leq r$, as shown in Lemma 106. This concludes the proof of claim (5.10).

We now consider the $(tn, 2^k, td)_2$ systematic code C_t obtained by repeating t times the code C. We remark that the value m can be thought of as the slope of the line $\mathrm{d}(C_t) \mapsto \mathrm{len}(C_t)$, and we proved that $m < \frac{g_2(k,2^r)}{2^r}$. Since $k \leq r$ we can apply Lemma 103, which ensures that $g_2(k, 2^{r+b}) = 2^b g_2(k, 2^r)$, namely the Griesmer bound computed on the powers of 2 is itself a line, and its slope is strictly greater than m. Due to this, we can find a pair $(t,\ b)$ such that the code C_t is an $(tn, 2^k, td)_2$ systematic code where

1. $td > 2^b$,

2. $tn < g_2(k, 2^b)$.

We can now apply Lemma 24 to C_t, and find a systematic code with length tn and distance equal to 2^b, which means we have an $(tn, k, 2^b)_2$ systematic code for which the length is $tn < g_2(k, 2^b)$. This however contradicts Corollary 102, hence for each $k \leq r$ we have

$$S_2(k, 2^r - 2^s) \geq g_2(k, 2^r - 2^s).$$

□

Corollary 108. *Let r and s be integers such that $r > s \geq 1$, and let d be either $2^s - 1$ or $2^r - 2^s - 1$. Then $S_2(k, d) \geq g_2(k, d)$.*

Proof. We prove it for $d = 2^r - 2^s - 1$, and the same argument can be applied to $d = 2^s - 1$ by applying Corollary 102 instead of Theorem 107.
Suppose by contradiction $S_2(k, d) < g_2(k, d)$, i..e. there exists an $(n, k, d)_2$ systematic code for which

$$n < g_2(k, d). \tag{5.13}$$

We can extend such a code to an $(n + 1, k, d + 1)_2$ systematic code C by adding a parity-check component to each codeword. Then C has distance $\mathrm{d}(C) = d + 1 = 2^r - 2^s$, so we can apply Theorem 107 to it, finding

$$n + 1 \geq g_2(k, d + 1).$$

Observe that d is odd, so applying Lemma 104 we obtain

$$n + 1 \geq g_2(k, d + 1) = g_2(k, d) + 1 \quad \Longrightarrow \quad n \geq g_2(k, d),$$

which contradicts (5.13). □

5.2 Versions of the Griesmer bound holding for nonlinear codes

In this section we collect some minor results which can be seen as bounds on the length of systematic codes, useful for a better understanding of the structure of such codes. An example of codes meeting these bounds are Simplex codes, while Preparata codes and Kerdock codes are close to these bounds. We will discuss some properties of Simplex codes in Section 5.5. We recall that Preparata codes are $\left(2^{2m}, 2^{2^{2m}-4m}, 6\right)_2$ systematic codes while Kerdock codes are $\left(2^{2m}, 2^{4m}, 2^{2m-1} - 2^{m-1}\right)_2$ systematic codes, both with $m \geq 2$. For $m = 2$ the two codes are both equivalent to the Nordstrom-Robinson code, which is a $(16, 2^8, 6)_2$ systematic binary code meeting the bound in Corollary 112.
In Table 5.2 there is a (not exhaustive) list of parameters n, d for which the binary bound in Equation (5.18) outperforms some known bounds, such as the Singleton Bound, the Elias bound, the Hamming Bound and the Johnson Bound.

5.2.1 Bound A

For systematic binary codes we can improve the Singleton bound as follows.

Proposition 109 (Bound A)**.**

$$S_2(k, d) \geq k + \left\lceil \frac{3}{2} d \right\rceil - 2.$$

Proof. We will proceed in a similar manner as in the proof of the Griesmer bound.
We consider a binary $(n = S_2(k, d), 2^k, d)_2$ systematic code C. We consider the set S of all codewords whose weight in their systematic part is 1. Let c be a codeword in this set with minimum weight:

$$\mathrm{w}(c) = \min_{x \in S} \{\mathrm{w}(x)\}. \tag{5.14}$$

Since we can always assume without loss of generality that the zero codeword belongs to C, the weight of c is at least d, and we denote it with $d+\Delta$, $\Delta \geq 0$. We also assume that the nonzero coordinates of c are the first $d+\Delta$, and that the first coordinate is the only nonzero systematic coordinate of c.

We construct a code C' by shortening C in the first coordinate and by puncturing it in the remaining $d+\Delta-1$ first coordinates. Since the shortening involves a systematic coordinate and the puncturing does not affect the systematic part of C, C' is an $(n-d-\Delta, 2^{k-1}, d')_2$ systematic code.

We consider now a codeword u in C', such that u has weight 1 in its systematic part. Then there exists a vector $v \in (\mathbb{F}_2)^{d+\Delta}$ such that the concatenation $(v \mid u)$ belongs to C. We remark that even though there may be many vectors satisfying this property, we can choose v such that its first component is 0, and this choice is unique. Therefore $(v \mid u) \in S$, and due to equation (5.14)

$$\mathrm{w}(v \mid u) = \mathrm{w}(v) + \mathrm{w}(u) \geq d + \Delta. \tag{5.15}$$

Moreover, we can also bound the distance of $(v \mid u)$ from c as follows:

$$\mathrm{d}(c, v \mid u) = d + \Delta - \mathrm{w}(v) + \mathrm{w}(u) \geq d \tag{5.16}$$

Summing together the inequalities (5.15) and (5.16) we have

$$d + \Delta + 2\mathrm{w}(u) \geq 2d + \Delta,$$

from which it follows that

$$\mathrm{w}(u) \geq \frac{d}{2}.$$

Since u has weight 1 in its systematic part, it means that its weight in the non-systematic part is at least $\frac{d}{2} - 1$. So u has $k-1$ systematic coordinates and at least $\frac{d}{2} - 1$ non-systematic coordinates:

$$\mathrm{len}(C') \geq (k-1) + \left(\frac{d}{2} - 1\right).$$

Since the length of C' is $n-d-\Delta$ we have

$$n-d-\Delta \geq k+\frac{d}{2}-2,$$

or equivalently

$$n \geq k+\frac{3d}{2}-2+\Delta$$

which implies the bound. □

5.2.2 Bound B

We derive from Proposition 101 a version of the Griesmer bound holding for any systematic code.

Remark 110. For any d, there exist $1 \leq r < q$ and $l \geq 0$ such that

$$q^l r \leq d < q^l(r+1) \leq q^{l+1} \tag{5.17}$$

Thus l has to be equal to $\lfloor \log_q d \rfloor$, and from inequality (5.17) we obtain $d/q^l - 1 < r \leq d/q^l$, namely $r = \lfloor d/q^l \rfloor$.

Corollary 111 (Bound B)**.** *Let* $l = \lfloor \log_q d \rfloor$ *and* $r = \lfloor d/q^l \rfloor$. *Then*

$$S_q(k,d) \geq d + \sum_{i=1}^{k-1} \left\lceil \frac{q^l r}{q^i} \right\rceil.$$

Proof. We denote $s = d - q^l r$. We remark that $s \leq n-k$, and so there are at least s non-systematic coordinates. With this notation, let C be an $(n, q^k, q^l r + s)_q$ systematic code. We build a new systematic code C_s by puncturing C in s non-systematic coordinates. C_s has parameters $(n-s, q^k, d_s)_q$, for a certain $q^l r \leq d_s \leq q^l r + s$.
If $q^l r \neq d_s$, we can apply Lemma 24, in order to obtain another code $\bar{C}$, so that we have an $(n-s, q^k, q^l r)_q$ systematic code. Due to Remark 110, it holds $1 \leq r < q$, so we can apply Proposition 101 to $\bar{C}$. We find $n - s \geq \sum_{i=0}^{k-1} \left\lceil \frac{q^l r}{q^i} \right\rceil$, hence $n \geq \sum_{i=0}^{k-1} \left\lceil \frac{q^l r}{q^i} \right\rceil + s$. We finally remark that for $i = 0$ we have $\left\lceil \frac{q^l r}{q^i} \right\rceil = q^l r$, and by adding s we obtain exactly d. So $n \geq d + \sum_{i=1}^{k-1} \left\lceil \frac{q^l r}{q^i} \right\rceil$. □

n	26	28	28	30	32	33
d	12	12	14	14	16	16
Elias bound	8	10	6	8	7	8
Bound B	7	9	5	7	6	7

Table 5.2: Bound B

We also derive a similar bound for binary codes, whose proof relies on Theorem 107 instead of Proposition 101.

Corollary 112 (Bound B, binary version)**.** *Let C be an $(n, 2^k, d)_2$ systematic code with d even. Let r and s be the smallest integers such that $2^r - 2^s \leq d < 2^r$, namely $r = \lceil \log_2(d+1) \rceil$ and $s = \lceil \log_2(2^r - d) \rceil$. Then*

$$n \geq d + \sum_{i=1}^{k-1} \left\lceil \frac{2^r - 2^s}{2^i} \right\rceil . \tag{5.18}$$

Proof. It follows directly from Theorem 107. □

In Table 5.2 we list some values n and d for which Bound B in Proposition 112 outperforms known bounds. The first two rows are respectively n and d. In the third row, we have the maximum combinatorial dimension allowed by the Elias Bound (EB). The last row is the bound obtained using Equation (5.18). We did not list other bounds in the table since for these values n and d the combinatorial dimensions obtained from the Hamming bound, the Singleton bound and the Johnson bound are at least equal to the one obtained from the Elias bound, while the Plotkin bound cannot be applied. More detailed tables for binary codes are in Appendix A.1.

5.2.3 Bound C

The following two bounds can be applied to nonlinear codes.

Proposition 113 (Bound C)**.** *Let l be the maximum integer such that q^l divides d, and let $h = \min(k-1, l)$. Then*

$$S_q(k,d) \geq N_q(q^k,d) \geq \sum_{i=0}^{h} \left\lceil \frac{d}{q^i} \right\rceil .$$

Proof. First, notice that $d = q^l r, \quad q \nmid r$. If $(k-1) \mid l$, we apply Lemma 100. Otherwise $h = l$, and d is not divisible for higher powers of q, and the last term of the sum is $\frac{d}{q^l}$. □

We remark that, if there exists an $(n, M, d)_q$ code, then there exists also an $(n, q^k, d)_q$ code, with $q^k \leq M$. By Proposition 113 we have

$$N_q(M,d) \geq \sum_{i=0}^{h} \left\lceil \frac{d}{q^i} \right\rceil .$$

5.3 Classification of optimal binary codes with 4 codewords

In the previous sections we have focused our attention on the distance, proving that for particular choices of d the length of optimal systematic codes is at least the Griesmer bound, for each possible dimension. In the next sections we deal with the task of characterize optimal systematic codes depending on their dimension. In particular in this section we prove that all optimal binary codes with 4 codewords are linear codes, and so they are systematic codes. We recall our convention $0 \in C$. A first version of this proof appeared in [15].

Lemma 114. $N_2(4,d) = S_2(2,d) = L_2(2,d)$.

Proof. Since $N_q(q^k,d) \leq L_q(k,d)$ for each choice of parameters, we are going to show that $N_2(4,d) \geq L_2(2,d)$ and this will conclude the proof.
Let $C = \{c_0, c_1, c_2, c_3\}$ be an optimal $(n,4,d)_2$ code, i.e. $n = N_2(4,d)$, and we assume without loss of generality that c_0 is the zero

codeword. The weights of c_1 and c_2 are at least d, and their distance is $d(c_1, c_2) = \mathrm{w}(c_1 + c_2) \geq d$. Therefore the linear code generated by c_1 and c_2 have the same minimum distance as C, and it follows that $n \geq L_2(2, d)$. □

A consequence of Lemma 114 is that the Griesmer bound holds for all binary (nonlinear) codes with 4 codewords. Furthermore, using the argument of the proof of Lemma 114 we can build (binary optimal) linear codes starting from nonlinear ones. This construction is however not necessary, as explained in the following theorem.

Theorem 115. *Let C be an optimal $(n, 4, d)_2$ code. Then C is a linear code.*

Proof. As in the proof of Lemma 114, we assume that c_0 is the zero codeword. If C is not linear, then there exists at least a position i for which the i-th coordinate of c_3 is different from the i-th coordinate of $c_1 + c_2$. Looking at the i-th components of the four codewords as a vector v in $(\mathbb{F}_2)^4$ we claim to have only two possibilities: either $\mathrm{w}(v) = 1$ or $\mathrm{w}(v) = 3$. In fact, $\mathrm{w}(v) = 0$ implies that C is not optimal, $\mathrm{w}(v) = 4$ contradicts the fact that $c_0 \in C$ and $\mathrm{w}(v) = 2$ contradicts the choice of i. Without loss of generality we can assume that we are in one of the following two cases:

$$v = (0, 0, 0, 1) \qquad \text{or} \qquad v = (0, 1, 1, 1)$$

We start from the first case, namely $\mathrm{w}(v) = 1$, and we consider the $[n, 2, d]_2$ linear code $\bar{C}$ generated by c_1 and c_2. Clearly, all codewords in $\bar{C}$ have the i-th component equal to zero. Then we can puncture $\bar{C}$, obtaining a $[n-1, 2, d]_2$ linear code, contradicting the fact that C is optimal.

We consider the second case, namely $\mathrm{w}(v) = 3$. We consider the code $\tilde{C}$ obtained by adding c_3 to each codeword in C. $\tilde{C}$ is an optimal code with the same parameters as C, and the zero codeword still

belongs to the code. However what we obtain looking at the i-th coordinate is a vector of weight 1, and we can use the same argument as in the first case. □

Corollary 116. *The Griesmer bound holds for binary codes with* 4 *codewords. Furthermore*

$$N_2(4,d) = S_2(4,d) = L_2(2,d) = \begin{cases} \frac{3}{2}d, & \text{if } d \text{ is even} \\ \frac{3}{2}(d+1)-1, & \text{if } d \text{ is odd} \end{cases}$$

Proof. The fact that the Griesmer bound holds for all codes of size 4 follows directly from Lemma 114 or Theorem 115. This implies that

$$N_2(4,d) \geq d + \left\lceil \frac{d}{2} \right\rceil$$

We consider d even, so that the previous equation is $N_2(4,d) = \frac{3}{2}d$. It is straightforward to exhibit a $\left[\frac{3}{2}d, 2, d\right]_2$ linear code C, and this concludes the proof in the case of d even. On the other hand, by puncturing C we obtain a $\left[\frac{3}{2}d-1, 2, d-1\right]_2$ linear code, which proves the case of odd distance. □

5.4 On the structure of optimal binary codes with 8 codewords

We consider in this section optimal codes with 8 codewords. First we prove that for these codes the Plotkin bound and the Griesmer bound coincide, implying that the Griesmer bound actually holds also for them.

Proposition 117. *For any d, $N_2(8,d) \geq g_2(3,d)$, namely*

$$N_2(8,d) \geq \begin{cases} 7h, & \text{if } d = 4h \\ 7h+3, & \text{if } d = 4h+1 \\ 7h+4, & \text{if } d = 4h+2 \\ 7h+6, & \text{if } d = 4h+3 \end{cases}. \tag{5.19}$$

Proof. Let us consider an $(N_2(8,d),8,d)_2$ code C. Let $h = \left\lfloor \frac{d}{4} \right\rfloor$. There are four cases for d:

$$d = 4h, \qquad d = 4h+1, \qquad d = 4h+2, \qquad d = 4h+3.$$

We start with the case $d = 4h$ (so $h \geq 1$), for which

$$g_2(3,4h) = \sum_{i=0}^{2} \left\lceil \frac{4h}{2^i} \right\rceil = 7h.$$

On the other hand, by the Plotkin bound we have

$$N_2(8,d) \geq \min \left\{ n \in \mathbb{N} \mid 8 \leq 2 \left\lfloor \frac{4h}{8h-n} \right\rfloor \right\}.$$

Assuming $n < 7h$, we have $8h - n > h$. This implies that

$$4 > \frac{4h}{8h-n},$$

which contradicts our hypothesis and shows that the Griesmer bound and the Plotkin bound coincide.

In the case of $d = 4h+2$,

$$g_2(3,4h+2) = \sum_{i=0}^{2} \left\lceil \frac{4h+2}{2^i} \right\rceil = (4h+2) + (2h+1) + (h+1) = 7h+4.$$

By the Plotkin bound

$$\frac{4h+2}{8h+4-N_2(8,d)}$$

which is equivalent to $N_2(8,d) \geq 7h+4$.

In the case of $d = 4h+1$,

$$8 \leq 2 \left\lfloor \frac{4h+2}{8h+3-N_2(8,d)} \right\rfloor,$$

hence $N_2(8,d) \geq 7h+3$.

Finally, in the case of $d = 4h+3$, by the same computation as above we obtain that $N_2(8,d) \geq 7h+6$. □

Theorem 118. *For any d, $L_2(3,d) = g_2(3,d)$.*

Proof. We consider the following three binary matrices:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; 1_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \; N_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

We remark that the code generated by I_3 (resp. $[\, I_3 \mid 1_3 \,]$ and $[\, I_3 \mid N_3 \,]$) is a $[3,3,1]_2$ (resp. a $[4,3,2]_2$ and a $[6,3,3]_2$) linear code. These codes meet the Griesmer bound. We denote with G_3 the matrix $[\, I_3 \mid N_3 \mid 1_3 \,]$, i.e.

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

The code generated by G_3 is a $[7,3,4]_2$ linear code, which again attains the Griesmer bound. Thus, $L_2(3,d) = g_2(3,d)$ for $1 \leq d \leq 4$.

Let $d = 4h$. We denote with $G_{3,h}$ the $3 \times 7h$ matrix obtained by repeating h times the matrix G_3. The code generated by $G_{3,h}$ is a $[7h,3,4h]_2$ linear code, which attains the Griesmer bound.

For the other three cases, we consider the matrices

$$\begin{cases} [\, G_{3,h} \mid I_3 \,] \\ \\ [\, G_{3,h} \mid I_3 \mid 1_3 \,] \\ \\ [\, G_{3,h} \mid I_3 \mid N_3 \,], \end{cases}$$

that generate, respectively, a $[7h+3,3,4h+1]_2$, a $[7h+4,3,4h+2]_2$ and a $[7h+6,3,4h+3]_2$ linear code, each attaining the Griesmer bound. □

Propositions 117 and Theorem 118 imply the following corollary.

Corollary 119. *For any d, $N_2(8,d) = S_2(3,d) = L_2(3,d)$, and*

$$N_2(8,d) = \begin{cases} 7h, & \text{if } d = 4h \\ 7h+3, & \text{if } d = 4h+1 \\ 7h+4, & \text{if } d = 4h+2 \\ 7h+6, & \text{if } d = 4h+3 \end{cases} \tag{5.20}$$

5.5 A family of optimal systematic codes

In previous sections we identified several sets of parameters for which the Griesmer bound holds in the systematic case. In this section we focus our attention on binary systematic (nonlinear) code for which the Griesmer bound does not hold. It is known that there exist pairs (k,d) for which $N_2(2^k,d) < g_2(k,d)$, but it has not been clear whether the same is true for systematic codes. In this section we construct a family of optimal systematic nonlinear codes contradicting the Griesmer bound. In Chapter 3 we have shown how Levenshtein proposed a method to construct optimal binary codes meeting the Plotkin bound, provided the existence of enough Hadamard matrices. In particular, given a Hadamard matrix of order 2^k+4, it is possible to construct a $(2^k+3, 2^k, 2^{k-1}+2)_2$ code D_k. We recall that binary codes attaining the Plotkin bound are equidistant codes.

Definition 120. A code C is called an *equidistant* code if any two codewords have the same distance d.

We consider now the family of binary simplex codes $\mathcal{S}_k$, which can be defined as the codes generated by the $k \times \left(2^k-1\right)$ matrices whose columns are all the nonzero vectors of $(\mathbb{F}_2)^k$. Simplex codes are $[2^k-1, k, 2^{k-1}]_2$ equidistant codes. The following proposition follows directly from the application of the Plotkin bound to codes with size 2^k and distance a multiple of 2^{k-1}.

Proposition 121. *Let $h \geq 1$ be a positive integer. Then*

$$N_2(2^k, 2^{k-1}h) \geq \left(2^k-1\right)h.$$

We recall that all $[(2^k-1)h, k, (2^{k-1})h]_2$ codes are equivalent to a sequence of Simplex codes [8]. This fact lead to the following corollary.

Corollary 122. *Let $h \geq 1$, then $N_2\left(2^k, 2^{k-1}h\right) = S_2\left(k, 2^{k-1}h\right) = L_2\left(k, 2^{k-1}h\right) = \left(2^k-1\right)h$.*

We now make use of D_k and $\mathcal{S}_k$ to construct our claimed family $\mathcal{C}_k$ of optimal systematic codes.
We consider $\mathcal{C}_k$ the $(2^{k+1}+2, 2^k, d)_2$ code, with the following properties:

- puncturing $\mathcal{C}_k$ in the last 2^k+3 coordinates we obtain $\mathcal{S}_k$;
- puncturing $\mathcal{C}_k$ in the first 2^k-1 coordinates we obtain D_k.

Note that such a code is completely defined. Since $\mathcal{S}_k$ is a linear code and both D_k and $\mathcal{S}_k$ are equidistant codes, $\mathcal{C}_k$ is an equidistant systematic code with distance $d = 2^k+2$.
Applying the Plotkin bound to these parameters, we can see that $\mathcal{C}_k$ is not an optimal code since it has only 2^k codewords instead of 2^k+2. However, if $k \geq 2$, it is optimal as a systematic code, since we can add to it at most two codewords and therefore we cannot increase its dimension while keeping the same distance. On the other hand, by the Griesmer bound we obtain

$$g_2(k, 2^k+2) = \sum_{i=0}^{k-1} \left\lceil \frac{2^k+2}{2^i} \right\rceil = \sum_{i=0}^{k-1} 2^{k-i} + \sum_{i=0}^{k-1} \left\lceil \frac{2}{2^i} \right\rceil .$$

By direct computation $g_2(k, 2^k+2) = 2^{k+1}+k-1$. Since $\text{len}(\mathcal{C}_k) = 2^{k+1}+2$, if $k > 3$ then $\mathcal{C}_k$ contradicts the Griesmer bound.

Proposition 123. *The family $\mathcal{C}_k$ is a family of optimal systematic equidistant binary codes.*

While in Sections 5.3 and 5.4 we have shown that codes of dimension 2 or 3 cannot contradict the Griesmer bound, by using the

family $\mathcal{C}_k$ we can obtain for each possible $k > 3$ an optimal systematic code whose length is smaller than the length of any possible linear code with the same dimension and distance, as stated in the following theorem.

Theorem 124. *Let $k > 3$. If there exists a Hadamard matrix of order $2^k + 4$, then there exists at least a distance d for which $S_2(k, d) < L_2(k, d)$.*

On the other hand, the family of optimal systematic codes presented in this section have distance $2^k + 2$. By puncturing them in a non-systematic component, for each $k > 3$, it is possible to construct $(2^{k+1} + 1, 2^k, 2^k + 1)_2$ optimal systematic codes contradicting the Griesmer bound. Theorem 107 and Corollary 108 imply that for $k < 3$ optimal systematic codes have to satisfy the Griesmer bound. Putting all together we can state the following theorem.

Theorem 125. *Let r be a positive integer, and let $d = 2^r + 1$ or $d = 2^r + 2$. Then*

1. *if $r < 3$ then all optimal systematic binary codes with dimension k and distance d have length at least equal to $g_2(k, d)$;*

2. *if $r > 3$, assuming there exists a Hadamard matrix of order $2^k + 4$, then $S_2(k, d) < L_2(k, d)$.*

This leaves as open problem the case $r = 3$, namely the case of a code whose distance is either 9 or 10.
Another open problem regarding systematic codes was presented by Bellini, Guerrini and Sala in [7]. In Section 2.10 we recalled the bound they proposed for $A_q^*(n, d)$, and we remarked that they pointed out how all known families of optimal codes were systematic or systematic embedding (Definition 58). In Chapter 3 we introduced Hadamard codes. We consider here in particular the Hadamard code obtained from H_{20}, which is a $(19, 20, 10)_2$ optimal code. Up to equivalence,

there exist three Hadamard codes with the same parameters. We denote them with $\mathcal{A}_{20}^{(1)}$, $\mathcal{A}_{20}^{(2)}$ and $\mathcal{A}_{20}^{(3)}$ and we show here their codebooks. It is possible to check that these three codes, as all codes which are equivalent to them, are not systematic embedding. Since these codes are obtained from the only three Hadamard matrices of order 20, it follows that $A_2^*(19,10) < A_2(19,10)$.

$$\mathcal{A}_{20}^{(1)} = \begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&1&0&0&1&1&1&1&0&1&0&1&0&0&0&0&1&1&0\\
0&0&1&0&0&1&1&1&1&0&1&0&1&0&0&0&0&1&1\\
1&0&0&1&0&0&1&1&1&1&0&1&0&1&0&0&0&0&1\\
1&1&0&0&1&0&0&1&1&1&1&0&1&0&1&0&0&0&0\\
0&1&1&0&0&1&0&0&1&1&1&1&0&1&0&1&0&0&0\\
0&0&1&1&0&0&1&0&0&1&1&1&1&0&1&0&1&0&0\\
0&0&0&1&1&0&0&1&0&0&1&1&1&1&0&1&0&1&0\\
0&0&0&0&1&1&0&0&1&0&0&1&1&1&1&0&1&0&1\\
1&0&0&0&0&1&1&0&0&1&0&0&1&1&1&1&0&1&0\\
0&1&0&0&0&0&1&1&0&0&1&0&0&1&1&1&1&0&1\\
1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&1&1&1&0\\
0&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&1&1&1\\
1&0&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&1&1\\
1&1&0&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&1\\
1&1&1&0&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1\\
1&1&1&1&0&1&0&1&0&0&0&0&1&1&0&0&1&0&0\\
0&1&1&1&1&0&1&0&1&0&0&0&0&1&1&0&0&1&0\\
0&0&1&1&1&1&0&1&0&1&0&0&0&0&1&1&0&0&1\\
1&0&0&1&1&1&1&0&1&0&1&0&0&0&0&1&1&0&0
\end{bmatrix}$$

$$\mathcal{A}_{20}^{(2)} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}$$

$$\mathcal{A}_{20}^{(3)} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1
\end{bmatrix}$$

Chapter 6

Entropy extractors and Codes

> *If numbers aren't beautiful, I don't know what is.*
>
> Paul Erdös

In this chapter we discuss entropy extractors, with a focus on linear compression functions. In Chapter 4 we introduced the notion of entropy extraction and we defined two well known post-processing functions, the Von Neumann procedure and binary linear extractors. In Section 6.1 we formalise the notion of non-binary Von Neumann procedure and discuss some properties of this extractor, while in Section 6.2 we analyse linear extractors over finite fields. Several RNGs already use digitisation procedures or post-processing functions similar to the ideas in this chapter. The intent of this work is not to propose new entropy extractors, our aim is to formally describe these techniques and the effects of their applications. Previous works on linear extractors mainly consider the binary case and provide bounds on the resulting entropy relying only on the minimum distance of the code generated by the matrix associated to the extractor. We obtain new bounds showing how the TVD from the uniform distribution of the output vectors is related not only to the minimum distance, but on the entire weight distribution of the code.

6.1 A generalisation of the Von Neumann procedure

Let X_t be an IID Random number generator, whose output is in a finite set Ω. Then the probability of X_t to output a certain element ω is independent on time, namely

$$\mathbb{P}(X_t = \omega) = \mu_x(\omega) \quad \forall \omega \in \Omega$$

The RNG X_t provides a certain entropy

$$H(X) = -\sum_{\omega \in \Omega} \mu_x(\omega) \log_{|\Omega|}(\mu_x(\omega)).$$

For cryptographic purposes we need bit sequences with maximum entropy, so we want an IID binary generator Y_s with

$$H(Y) = 1.$$

This is achieved when $\Omega_y = \mathbb{F}_2 = \{0, 1\}$, and

$$\mathbb{P}(Y_s = 0) = \mathbb{P}(Y_s = 1) = \frac{1}{2} \quad \forall s$$

Given a certain IID RNG (X_t, Ω, μ_x) we want to build Y_s satisfying the properties described above.
We consider non-overlapping output pairs from X_t, i.e. we consider the process

$$(X_1, X_2), (X_3, X_4), \ldots, (X_{2t-1}, X_{2t}), \ldots$$

so that we have a process $(X_{2t-1}, X_{2t})_{t>1}$, whose output is in the space $\Omega \times \Omega$. We remark that X_t is assumed to be IID, hence this new process is also IID. We also observe that each finite set Ω can be represented as the set $\mathbb{Z}_n$, with $n = |\Omega|$.
We consider Algorithm 2. The output sequence Y_s is the desired binary sequence, while Z_q is a sequence which contains a representative of each pair (X_{2t-1}, X_{2t}) with $X_{2t-1} = X_{2t}$.

Example 3. $N_0 = 6$, $X_t \in \mathbb{Z}_4$ with output $(0, 2, 3, 3, 3, 2)$.
First of all, we consider disjoint pairs from the input sequence, namely $((0, 2), (3, 3), (3, 2))$, and we set $t = s = q = 1$.

$t = 1$ we consider the first pair $(0, 2)$. We observe that $0 < 2$, hence $Y_1 = 0$, and we increment s and t.

$t = 2$ we check the second pair $(3, 3)$. They are equal, so $Z_q = Z_1 = 3$ and we increment both t and q.

$t = 3$ we look at the last pair $(3, 2)$. $3 \neq 2$, so $Y_2 = 1$, and we increment both t and s.

$t = 4$ $t > N$, so we stop.

The output of the algorithm are the two sequences $(Y_s)_{s\in\{1,2\}} = (0, 1)$ and $(Z_q)_{q\in\{1\}} = (3)$.

Theorem 126. *If X_t is IID then Y_s is IID and*

$$\mathbb{P}(Y_s = 0) = \mathbb{P}(Y_s = 1) = \frac{1}{2} \quad \forall s.$$

ALGORITHM 2:

Data: $X_T \in \mathbb{Z}_\kappa$, $T \in [1, N_0]$

Result: Y_s, Z_q

Initialization:

$N_1 = \lfloor \frac{N_0}{2} \rfloor$;

Build the process $(X_{2t-1}, X_{2t})_{t\in[1,N_1]}$;

$t = 1$;

$s = 1$;

$q = 1$;

while $t \leq N_1$ **do**

  if $(X_{2t-1} == X_{2t})$ **then**

    $Z_q = X_{2t-1}$;

    $q = q + 1$;

  else

    if $X_{2t-1} < X_{2t}$ **then**

      $Y_s = 0$;

    else

      $Y_s = 1$;

    end

    s=s+1

  end

  $t = t + 1$;

end

Algorithm 2: A generalisation of the Neumann procedure to non-binary random variables.

Proof. $\mathbb{P}(Y_s = 0) = \mathbb{P}(X_{2t-1} < X_{2t})$, but X_t is an IID process, hence $\mathbb{P}(X_{2t-1} < X_{2t}) = \mathbb{P}(X_{2t-1} > X_{2t})$. □

Corollary 127. *If X_t is IID then $H(Y) = 1$*

Proof. Follows from the previous Theorem and the definition of entropy. □

Theorem 128. *The expected length of the sequence Y_s is given by $\left\lfloor \frac{N_0}{2} \right\rfloor \left(1 - \|\mu_x\|_2^2\right)$, where N_0 is the length of the input sequence, and μ_x is the probability mass function of X_T.*

Proof. Given a pair (X_{2t-1}, X_{2t}), the probability of not outputting anything is equal to the probability of X_{2t-1} to be equal to X_{2t}. It holds

$$\mathbb{P}(X_{2t-1} = X_{2t}) = \sum_{\alpha} \mathbb{P}(X_{2t-1} = \alpha | X_{2t} = \alpha)\mathbb{P}(X_{2t} = \alpha),$$

but X_t is IID, hence

$$\mathbb{P}(X_{2t-1} = X_{2t}) = \sum_{\alpha} \mu_x(\alpha)^2 = \|\mu_x\|_2^2. \tag{6.1}$$

The probability to output a bit from a given pair is therefore $1 - \|\mu_x\|_2^2$. □

Remark 129. The expected length of Y_s is maximized when $\|\mu_x\|_2^2$ is minimized, i.e. when μ_x is uniform. It follows that the expected length of Y_s is always at most $\frac{n-1}{n} \left\lfloor \frac{N_0}{2} \right\rfloor$.

Proposition 130. *The expected length of the sequence Z_q is given by $\left\lfloor \frac{N_0}{2} \right\rfloor \|\mu_x\|_2^2$.*

We can say something more on the process Z_q. First of all, due to the hypothesis of independence for X_t, also Z_q is IID. Furthermore, we can obtain its probability mass function, as shown in Proposition 131.

Proposition 131. *The probability mass function of Z_q is given by*

$$\mu_z(\alpha) = \frac{\mu_x(\alpha)^2}{\|\mu_x\|_2^2}. \tag{6.2}$$

Proof. $\mathbb{P}(Z_q = \alpha) = \mathbb{P}(X_{2t-1} = X_{2t} = \alpha)$, so

$$\mathbb{P}(Z_q = \alpha) = \frac{\mathbb{P}(X_{2t-1} = X_{2t} | X_{2t-1} = \alpha)\mathbb{P}(X_{2t-1} = \alpha)}{\mathbb{P}(X_{2t-1} = X_{2t})}.$$

X_t is an IID process, hence

$$\mathbb{P}(X_{2t-1} = X_{2t} | X_{2t-1} = \alpha) = \mathbb{P}(X_{2t} = \alpha) = \mathbb{P}(X_{2t-1} = \alpha) = \mu_x(\alpha).$$

We obtain

$$\mathbb{P}(Z_q = \alpha) = \frac{\mu_x(\alpha)^2}{\mathbb{P}(X_{2t-1} = X_{2t})}, \tag{6.3}$$

and using (6.1) we conclude. □

If we now reiterate the procedure using as input the sequence Z_q, the algorithm will give us as output two new sequences Y_s' and Z_q' . Being Z_q IID, we can apply previous theorems, obtaining

- Y_s' is IID and $H(Y_s') = 1$. The expected length of Y_s' is

$$\left\lfloor \frac{M}{2} \right\rfloor \left(1 - \|\mu_z\|_2^2\right),$$

  with M the length of Z_q and μ_z as in (6.2).

- Z_q' is IID, and its pmf is given by (6.2), using μ_z instead of μ_x.

Then, given X_t and applying the extractor two times, first on X_t and then to Z_q, we obtain a sequence of bits with no bias, and an expected length of

$$\left\lfloor \frac{N_0}{2} \right\rfloor \left(1 - \|\mu_x\|_2^2\right) + \left\lfloor \frac{M}{2} \right\rfloor \left(1 - \|\mu_z\|_2^2\right). \tag{6.4}$$

The expected length of Z_q was $M = \left\lfloor \frac{N_0}{2} \right\rfloor \|\mu_x\|_2^2$, while μ_z is as in (6.2). Using these informations, (6.4) becomes

$$\left\lfloor \frac{N_0}{2} \right\rfloor \left(1 - \|\mu_x\|_2^2\right) + \left\lfloor \frac{\left\lfloor \frac{N_0}{2} \right\rfloor \|\mu_x\|_2^2}{2} \right\rfloor \left(1 - \left\| \frac{\mu_x^2}{\|\mu_x\|_2^2} \right\|_2^2\right).$$

Suppose now we want to keep on reiterating the algorithm using as input the sequence Z. We consider as first input a sequence of length N_0 and pmf μ_0. We iterate the algorithm J times, and we indicate with $j = 1, \dots, J$ the iteration step.

For each j we have two output sequences, Y_j and Z_j, and we call N_j the expected length of Z_j, μ_j its pmf, and L_j the expected length of Y_j. Using Theorem 128, Proposition 130 and Proposition 131, we obtain:

$$\begin{cases} N_j = \frac{1}{2} N_{j-1} \left\| \mu_{j-1} \right\|_2^2 \\ \mu_j = \frac{\mu_{j-1}^2}{\left\| \mu_{j-1} \right\|_2^2} \\ L_j = \frac{1}{2} N_{j-1} \left(1 - \left\| \mu_{j-1} \right\|_2^2 \right). \end{cases} \tag{6.5}$$

Lemma 132. *The probability mass functions μ_j is such that*

$$\left\| \mu_j \right\|_2^2 = \left(\frac{\left\| \mu_0 \right\|_{2^{j+1}}}{\left\| \mu_0 \right\|_{2^j}} \right)^{2^{j+1}}.$$

Proof. We use induction, using equation (6.5) with initial step given by Proposition 131. □

Lemma 133. *The expected length N_j is given by*

$$N_j = \frac{N_0}{2^j} \cdot \frac{\left\| \mu_0 \right\|_{2^j}^{2^{j+1}}}{\prod_{i=0}^{j} \left\| \mu_0 \right\|_{2^i}^{2^i}}.$$

Proof. From equation (6.5) we can use induction, with initial step given by Proposition 130. We remark that, for $i = 0$, at the denominator we obtain the term $\left\| \mu_0 \right\|_1^1 = 1$, and that for $i = j$ we get instead $\left\| \mu_0 \right\|_{2^j}^{2^j}$, which can be simplified together with the numerator. □

Theorem 134. *Padding together all the output sequences Y_j, we obtain a binary sequence with no bias, whose expected length $\bar{L}_J$ is given by*

$$\bar{L}_J = N_0 \cdot \left[\sum_{j=0}^{J} \frac{1}{2^{j+1}} \frac{1}{\prod_{i=0}^{j} \left\| \mu_0 \right\|_{2^i}^{2^i}} \left(\left\| \mu_0 \right\|_{2^j}^{2^{j+1}} - \left\| \mu_0 \right\|_{2^{j+1}}^{2^{j+1}} \right) \right] =: N_0 \cdot R_J \tag{6.6}$$

Proof. $\bar{L}_J$ is the sum of all L_j's, for $j = 0, \dots, J$. From equation (6.5) we see that L_j depends on N_{j-1} and on μ_{j-1}, so we can use Lemmas 132 and 133 to substitute the values. A few computational steps conclude the proof. □

Theorem 135. *The output/input bitrate ratio R_J converges for $J \to +\infty$.*

Proof. Suppose the term

$$A_j := \frac{1}{\prod_{i=0}^{j} \|\mu_0\|_{2^i}^{2^i}} \left(\|\mu_0\|_{2^j}^{2^{j+1}} - \|\mu_0\|_{2^{j+1}}^{2^{j+1}} \right)$$

is less than 1 for each j. Then $R_J = \sum_{j=0}^{J} \frac{1}{2^{j+1}} \cdot A_j$ is convergent for $j \to +\infty$. We observe that we can rewrite A_j as

$$A_j = \frac{\|\mu_0\|_{2^j}^{2^{j+1}}}{\prod_{i=0}^{j} \|\mu_0\|_{2^i}^{2^i}} \cdot \left(1 - \frac{\|\mu_0\|_{2^{j+1}}^{2^{j+1}}}{\|\mu_0\|_{2^j}^{2^{j+1}}} \right) =: B_j \cdot C_j$$

Remark 136. Given $p > r$, $\|\cdot\|_p \leq \|\cdot\|_r$. Hence using $p = 2^{j+1}$ and $r = 2^j$, we can prove that $C_j < 1$.

If we prove that also $B_j < 1$ we can conclude. First of all, we have

$$B_j < 1 \quad \iff \quad \|\mu_0\|_{2^j}^{2^{j+1}} < \prod_{i=0}^{j} \|\mu_0\|_{2^i}^{2^i}$$

Taking the logarithm on both sides of the inequality, we obtain the claim

$$2^{j+1} \log \|\mu_0\|_{2^j} < \sum_{i=0}^{j} 2^i \log \|\mu_0\|_{2^i}$$

We observe that for the term on the right-hand side holds

$$\sum_{i=0}^{j} 2^i \log \|\mu_0\|_{2^i} > \sum_{i=0}^{j} \log \|\mu_0\|_{2^j} = (j+1) \log \|\mu_0\|_{2^j}$$

and we obtain

$$2^{j+1} \log \|\mu_0\|_{2^j} < (j+1) \log \|\mu_0\|_{2^j}.$$

Due to $\|\mu_0\|_{2^j} < 1$, this last inequality is true if

$$2^{j+1} > j + 1,$$

hence we can conclude that $B_j < 1$, and therefore R_j converges. □

Corollary 137. *The ratio* $R = \lim_{J \to +\infty} R_j$ *is less than* $\frac{1}{2}$.

Proof. Follows from proof of Theorem 135. □

Using the same argument as in Remark 129, R is maximized if μ_0 is uniform.

Corollary 138. *The ratio* R *is bounded by* $\frac{n-1}{2n-1}$

Proof. It follows from Theorem 134 by using the uniform distribution instead of μ_0. □

6.2 New bounds for linear extractors

We show the connection between the Walsh spectrum of the output of a binary random number generator (RNG) and the bias of individual bits, and use this to show how previously known bounds on the performance of linear binary codes as entropy extractors can be derived by considering generator matrices as a selector of a subset of that spectrum. We explicitly show the connection with the code's weight distribution, then extend this framework to the case of non-binary finite fields by the Fourier transform. The results presented in this section can also be found in [48].

Our objective is to obtain sharp bounds for a finite number of iterations of a conditioning procedure applied to the output of an entropy source, rather than results relying on an asymptotic convergence or ones based on randomly choosing a conditioning function from among a large class, e.g. universal hash functions. We follow the recommendations set out by NIST [2] for the precise meaning to

be given to these terms. This is closely related to the subject of randomness extractors, as summarised for instance in [43]. We consider linear transformations applied to entropy sources producing independent output; bounds on the distance from the uniform distribution of such sequences have been shown in [28], [29] and [52], where the entropy source was assumed to produce biased independent bits, and the conditioning function was the generator matrix of a binary linear code.

We are especially interested in random variables that, in addition to satisfying said constraints, are discrete - in the sense that the number of possible outcomes is finite and the variables admit a discrete probability mass function $\mu_X(j) = \mathbb{P}(X = x_j)$; moreover, we begin by considering these variables to take values in a finite field $\mathbb{F}_p$ or a vector space $(\mathbb{F}_p)^k$, with particular regard to the special case of binary variables, $p = 2$.

In Section 6.2.1 we show the connection between the Walsh spectrum of the output of a binary random number generator (RNG) and the bias of individual bits, and use this in Section 6.2.2 to show how previously known bounds on the performance of linear binary codes as RNG post-processing functions can be derived as a special case by considering generator matrices as a selector of a subset of that spectrum, explicitly showing the connection with the code's weight distribution. We then extend this framework to the case of output in non-binary finite fields by use of the Fourier transform in Section 6.2.4.

6.2.1 Total variation distance and the Walsh-Hadamard transform

We show in the following one way in which the Walsh-Hadamard transform may be used to bound the total variation distance of binary random variables with a known probability mass function. This may

seem an unnecessary exercise since the TVD can simply be computed exactly from this knowledge, but aside from revealing some interesting structure to the calculation it will become more explicitly useful in the following section.

Consider a random vector $Y \in (\mathbb{F}_2)^k$ with probability mass function $\mu_Y(a) = \mathbb{P}(Y = \mathbf{a})$, where in writing a and $\mathbf{a}$ we use the binary representation of integers $a \in \mathbb{Z}_{2^k}$ as vectors $\mathbf{a} \in (\mathbb{F}_2)^n$.

$$\mathbf{a} = \left\{ a_j \,\middle|\, a = \sum_{j=0}^{k-1} a_j 2^j \right\} \in (\mathbb{F}_2)^k .$$

The j-th order Walsh function evaluated at a is

$$h_j(a) = (-1)^{\mathbf{j}\cdot\mathbf{a}} \tag{6.7}$$

with $\cdot$ the dot product on $(\mathbb{F}_2)^k$. The j^{th} Walsh characteristic function of Y as defined in [37] is

$$\chi_j(Y) = \sum_{a=0}^{2^k-1} h_j(a)\mu_Y(a) \tag{6.8}$$

$$= \mathbb{E}[h_j(Y)] \tag{6.9}$$

Note that the dot product of two binary vectors $\mathbf{b}\cdot\mathbf{v}$ is the bitwise sum, i.e. the linear combination, of those elements $\mathbf{v}(i)$ that correspond to the nonzero entries of $\mathbf{b}$; therefore, the random variable

$$h_b(Y) = (-1)^{\mathbf{b}\cdot Y}$$

will take value -1 if the linear combination of the selected elements of Y is equal to one, and 1 otherwise. The sum of the selected elements is itself a random variable, $B = \mathbf{b}\cdot Y$ following the Bernoulli distribution with probability $\mu_B(1)$ of being equal to 1; it follows that

$$h_b(Y) = 1 - 2B,$$

and hence we can conclude that

$$\begin{aligned}\chi_b(Y) &= \mathbb{E}[h_b(Y)] \\ &= 1 - 2\mu_B(1).\end{aligned}$$

We recall that the bias of a Bernoulli random variable $B \in \mathbb{F}_2$ is commonly defined as

$$\begin{aligned}\frac{\varepsilon_B}{2} &= \frac{1}{2}\left|\mathbb{P}(B=1) - \mathbb{P}(B=0)\right| \\ &= \frac{1}{2}\left|2\mathbb{P}(B=1) - 1\right| \\ &= \frac{1}{2}\left|2\mathbb{E}[B] - 1\right|,\end{aligned}$$

and we observe that the Walsh characteristic of $\mathbf{b}\cdot Y$ leads to the bias of the b^{th} linear combination of the elements of Y via the relation

$$|\chi_b(Y)| = \varepsilon_{\mathbf{b}\cdot Y}\,. \tag{6.10}$$

In particular, the combinations corresponding to exact powers of two, $b = 2^j$, lead to the bias of each individual element of Y; and the zeroth Walsh characteristic, corresponding to $b = 0$, will be equal to 1 in all entries, in all cases.

The set $\{\, h_i \,\}$ is known to correspond to the rows of a Hadamard matrix H of size 2^k; the set of all Walsh characteristics of Y can thus be written compactly in matrix notation as

$$\chi(Y) = H\mu_Y\,. \tag{6.11}$$

As a matter of notation, for a uniformly distributed random variable $U \in (\mathbb{F}_2)^k$ we have

$$\begin{aligned}\mu_U &= \frac{\mathbf{1}}{2^k} \\ \chi(U) &= \mathbb{I}_{2^k}(\cdot, 1),\end{aligned}$$

with $\mathbf{1}$ a column vector of ones and $\mathbb{I}_{2^k}(\cdot, 1)$ the first column of the identity matrix of size 2^k. We may use this to estimate the total variation distance of Y from uniform as follows.

Theorem 139. *The total variation distance of a random vector $Y \in (\mathbb{F}_2)^k$ from uniform U,*

$$\begin{aligned}\mathrm{TVD}(Y,\mathcal{U}) &= \frac{1}{2}\left\|\mu_Y - \mu_U\right\|_1 \\ &= \frac{1}{2}\delta_Y\end{aligned}$$

is bounded by the sum of the bias of all non-trivial linear combinations of the output bits,

$$\delta_Y \leq \sum_{\mathbf{b}\in(\mathbb{F}_2)^k\setminus\mathbf{0}} \varepsilon_{\mathbf{b}\cdot Y}\,.$$

Proof.

$$\begin{aligned}\left\|\mu_Y - \mu_U\right\|_2 &= \left\|\frac{H^T H}{2^k}\left(\mu_Y - \mu_U\right)\right\|_2 && (6.12)\\ &= \frac{1}{2^{k/2}}\left\|\frac{H^T}{2^{k/2}}\left(\chi(Y) - \chi(U)\right)\right\|_2 && (6.13)\\ &\leq \left\|\chi(Y) - \chi(U)\right\|_1 && (6.14)\\ &= \sum_{\mathbf{b}\in(\mathbb{F}_2)^k\setminus\mathbf{0}} \varepsilon_{\mathbf{b}\cdot Y}. && (6.15)\end{aligned}$$

Here H^T is the transpose of the Hadamard matrix H. Equation (6.12) follows from $H^T/\sqrt{2^k}$ being the unitary inverse Hadamard transform; Equation (6.13) uses the definition of $\chi(Y)$ in Equation (6.11); and lastly, the bound in Equation (6.14) stems from the ℓ_1 bound on q-dimensional vector spaces, $\|\cdot\|_1 \leq q^{1/2}\|\cdot\|_2$. □

Corollary 140. *If the bits $Y(j) \in \mathbb{F}_2$ are i.i.d. with known bias ε_y, then*

$$\begin{aligned}\delta_Y &\leq \sum_{l=1}^{k} A_l \varepsilon_y^l \\ &= \sum_{l=1}^{k} \binom{k}{l} \varepsilon_y^l\end{aligned}$$

where A_l is the number of $\mathbf{b} \in (\mathbb{F}_2)^k$ with Hamming weight $\mathrm{w}(\mathbf{b}) = l$.

6.2.2 W-H bound on binary generator matrices as extractors

We now consider the previous bound as applied to random variables $Y = GX$, with $X \in (\mathbb{F}_2)^n$ a sequence of n Bernoulli random variables with known probability mass $\mathbb{P}(X = b) = \mu_X(b)$ and identical bias $\varepsilon_X = |1 - 2\mathbb{P}(X(i) = 1)|$ for each bit, and G the generator matrix of an $[n, k, d]_2$ linear code C with weight distribution $\{ A_l \}$. We recall that in the following sections all vectors are column vectors.

The definition of Walsh characteristic functions as expected values in Equation (6.9) directly leads to

$$\begin{aligned}\chi_b(GX) &= \mathbb{E}[h_b(GX)] \\ &= \sum_{x=0}^{2^n-1} (-1)^{\mathbf{b}\cdot G\mathbf{x}} \mu_X(x)\end{aligned}$$

We note here that the dot product in the Walsh function can equivalently be expressed using the transpose $\mathbf{b}^T$ as

$$\mathbf{b} \cdot G\mathbf{x} = \mathbf{b}^T G\mathbf{x}\,,$$

and in particular the product

$$\mathbf{c}^T = \mathbf{b}^T G$$

is a linear combination of the rows of G: since G is the generator matrix of a linear code C the rows of G form a basis of C, and hence any linear combination of them is again a word $\mathbf{c} \in C$. Consequently, just as the Walsh characteristic led to a measure of bias in Equation (6.10), we can conclude that

$$|\chi_b(Y)| = \varepsilon_{\mathbf{c}\cdot X}\,. \tag{6.16}$$

In other words, the bias of the b^{th} element of Y is equal to the bias of a linear combination of $\mathrm{w}(\mathbf{c})$ bits of X (compare to Equation (6.10)). This leads directly to the following bound.

Theorem 141. *Let $Y = GX$, with $X \in (\mathbb{F}_2)^n$ a sequence of n independent but not necessarily identically distributed Bernoulli random variables, and G the generator matrix of an $[n, k, d]_2$ linear code C. The total variation distance of the random variable $Y \in (\mathbb{F}_2)^k$ from uniform,*

$$\mathrm{TVD}(Y,\mathcal{U}) = \frac{\delta_Y}{2}$$

is bounded by the sum of the bias of all linear combinations of the bits in X defined by the codewords of C, in the following measure:

$$\begin{aligned}\delta_Y &\leq \sum_{\mathbf{b}\in(\mathbb{F}_2)^k\setminus\mathbf{0}} \varepsilon_{\mathbf{b}^T G\cdot X} \\ &= \sum_{\mathbf{c}\in C\setminus\mathbf{0}} \varepsilon_{\mathbf{c}\cdot X}\end{aligned}$$

Corollary 142. *If the bits $X(j) \in \mathbb{F}_2$ are i.i.d. with known bias ε_x, then*

$$\delta_Y \leq \sum_{l=d}^{n} A_l \varepsilon_x^l \,.$$

with $\{ A_l \}$ the weight distribution of C.

Note that Corollary 140 is closely related to Corollary 142 if we consider that in this context the $\{ A_l \}$ in the former correspond precisely to the weight distribution of the trivial code given by the message space itself, $(\mathbb{F}_2)^k$. A particular case of Corollary 142 for strictly binomial $\{ A_l \}$ was proved in [52], Theorem 6. We can thus recover the following known bound, already presented in Theorem 93 and whose original proof is in [28], Theorem 1.

Corollary 143. *Considering only the minimum distance d rather than the full weight distribution,*

$$\delta_Y \leq (2^k - 1)\varepsilon_x^d \,.$$

6.2.3 Total variation distance and the Fourier transform

The Hadamard transform is a special case of the Fourier transform constructed with primitive 2-nd root of unity $\omega_p = -1$. Employing the Fourier transform is natural in this setting and closely follows well-established techniques for the sum of continuous random variables, which have their own convolution theorem and proofs of convergence to a limiting distribution.

Given an integer $a \in \mathbb{Z}_{p^k}$, we denote its p-ary representation by

$$\mathbf{a} = \left\{ a_j \,\middle|\, a = \sum_{j=0}^{k-1} a_j p^j \right\} \in (\mathbb{F}_p)^k \ .$$

We shall use this notation interchangeably in the following as a natural indexing of the elements of $(\mathbb{F}_p)^k$.

Consider a random variable $Z \in \mathbb{F}_p$ with probability mass function

$$\mu_Z(j) = \mathbb{P}(Z = \mathbf{j}) \, .$$

Note that this implicitly assumes an ordering of the mass function μ_Z corresponding to the representation of elements $\beta_j \in \mathbb{F}_p$ as integers. The discrete Fourier transform of μ_Z may then be written in matrix form as

$$F_p \mu_Z = \lambda_Z \, ,$$

where λ is the set of eigenvalues of the circulant matrix generated by μ_Z. Indeed, the above can be restated in terms of the unitary DFT,

$$\hat{F}_p = \frac{F_p}{\sqrt{p}} \qquad \hat{F}_p^* = \frac{F_p^*}{\sqrt{p}}$$

with F_p^* the conjugate transpose of F_p, diagonalising the circulant matrix C_Z generated by μ_Z:

$$C_Z = circ(\mu_Z)$$
$$\hat{F}_p C_Z \hat{F}_p^* = \Lambda_Z$$

with Λ_Z the $p \times p$ diagonal matrix containing all eigenvalues of C_Z. Note that for a uniformly distributed random variable $U \in \mathbb{F}_p$, we have

$$\mu_U = \frac{\mathbf{1}}{p}$$
$$\lambda_U = F_p \mu_U = \mathbb{I}_p(\cdot, 1)$$

where $\mathbf{1}$ is a vector of ones of length p, and the only nonzero eigenvalue is the zeroth one, so the full set λ_U corresponds to the first column of the identity matrix of size p.

Lemma 144. *The mass function μ_Z of a random variable $Z \in \mathbb{F}_p$ satisfies*

$$\|\mu_Z - \mu_U\|_2 = \frac{1}{\sqrt{p}} \|\lambda_Z - \lambda_U\|_2$$

where $\lambda_Z = F_p \mu_Z$ is the discrete Fourier transform of Z.

Proof.

$$\|\mu_Z - \mu_U\|_2 = \left\| \frac{\hat{F}^*}{\sqrt{p}} (\lambda_Z - \lambda_U) \right\|_2 \tag{6.17}$$
$$= \frac{1}{\sqrt{p}} \|\lambda_Z - \lambda_U\|_2 \tag{6.18}$$
$$= \frac{1}{\sqrt{p}} \left(\sum_{j=1}^{p-1} (\lambda_Z(j))^2 \right)^{1/2} \tag{6.19}$$

Equation (6.18) follows from the unitary Fourier transform preserving ℓ_2 distance. □

We can obtain a first, crude bound on the TVD by considering the largest non-trivial eigenvalue, defined as follows for future reference.

Definition 145. Given a random variable $Z \in \mathbb{F}_p$ with mass function μ_Z and eigenvalues $F_p \mu_Z = \lambda_Z$, denote the greatest non-trivial eigenvalue by

$$\lambda_{Z*} = \max_{1 \le j \le p-1} |\lambda_Z(j)|.$$

Theorem 146. *The total variation distance of a random variable $Z \in \mathbb{F}_p$ from uniform,*

$$\mathrm{TVD}(Z,\mathcal{U}) = \frac{\delta_Z}{2}$$

is bounded by

$$\delta_Z \leq (p-1)^{1/2}\,\lambda_{Z*}$$

with λ_{Z*} as in Definition 145.

Proof. This follows from considering the worst-case scenario in which all eigenvalues λ_Z in Lemma 144, except the zeroth eigenvalue $\lambda_Z(0) = 1$, are equal to the greatest non-trivial eigenvalue λ_{Z*} by applying the bound on p-dimensional vector spaces $\|x\|_1 \leq p^{1/2}\|x\|_2$.

□

We can now consider how this affects the distribution of a sum of two variables, $S_2 = X_0 + X_1 \in \mathbb{F}_p$, which is the discrete convolution of the two probability masses,

$$\begin{aligned}\mathbb{P}(S_2 = r) &= \sum_{j\in\mathbb{F}_p} \mathbb{P}(Z_0 = j)\mathbb{P}(Z_1 = r-j) \\ &= \sum_{j=0}^{p-1} \mu_{Z_1}(r-j)\mu_{Z_0}(j).\end{aligned}$$

The distribution of S_2 may then be expressed in matrix notation as

$$\mu_{S_2} = C_{Z_1}\mu_{Z_0}\,, \tag{6.20}$$

where C_{Z_1} is the circulant matrix uniquely defined by μ_{Z_1}. In other words, the entry r, j of C_{Z_1} contains the measure under Z_1 of the element $\beta_r - \beta_j \in \mathbb{F}_p$, which we denote in matrix form by

$$\begin{aligned}C_{Z_1} &= \mu_{Z_1}(B)\,, \\ B(r,j) &= \beta_r - \beta_j\,.\end{aligned}$$

Considering the particular case of summing two identical variables Z with probability mass function μ_Z, the distribution of $S_2 = Z + Z$ may be written as

$$S_2 \sim C_Z \mu_Z,$$

where C_Z is the circulant matrix defined uniquely by μ_Z itself. By induction,

$$\begin{aligned} S_n &\sim C_Z^{n-1} \mu_Z \\ &\sim \left(\prod_{j=1}^{n-1} \frac{F_p^*}{p} \Lambda_Z F_p \right) \mu_Z \\ &\sim \frac{\hat{F}_p^*}{\sqrt{p}} \Lambda_Z^{n-1} F_p \mu_Z \\ &\sim \frac{1}{\sqrt{p}} \hat{F}_p^* \lambda_Z^n \,. \end{aligned} \tag{6.21}$$

As well as being conceptually equivalent to using the convolution theorem, this may also be seen as considering S_n as a Markov chain

$$\begin{aligned} S_0 &= Z \\ S_j &= S_{j-1} + Z \end{aligned}$$

with transition matrix C_Z.

Lemma 144 may be extended as follows.

Lemma 147. *The probability mass function μ_{S_n} of a random variable $S_n = \sum_{j=1}^n Z$, $Z \in \mathbb{F}_p$ satisfies*

$$\| \mu_{S_n} - \mu_U \|_2 = \left(\frac{p-1}{p} \right)^{1/2} \| \left(\lambda_{S_n} - \lambda_U \right)^n \|_2 \,.$$

Proof. The proof is substantially the same as that of Lemma 144, using Equation (6.21).

□

Lemma 148. *The total variation distance of* $S_n = \sum_{j=1}^{n} Z$, $Z \in \mathbb{F}_p$ *from uniform may be bounded by*

$$\delta_{S_n} \leq (p-1)^{1/2} \lambda_{Z*}^n \tag{6.22}$$

with λ_{Z*} *as in Definition 145.*

Proof. The proof follows by applying Lemma 147 in the same way as Lemma 144 was applied to Theorem 146, i.e. assuming each of the $p-1$ eigenvalues in Lemma 147 that are of magnitude less than 1 to be bounded by λ_*^n, using Equation (6.21).

□

Lemma 148 is a slight improvement on a known bound on the convergence rates of Markov chains on Abelian groups; see e.g. [41] Fact 7.

We have so far assumed an ordering of the mass function μ_X of a random variable $X \in \mathbb{F}_p$ according to the representation of the elements of $\mathbb{F}_p$ as integers. Similarly for vector spaces $X \in (\mathbb{F}_p)^k$ we may assume an ordering of μ_X by least significant digit. Generalising to the distribution of the sum S_2 of two random variables $X_0, X_1 \in (\mathbb{F}_p)^k$, this may still be expressed in a form such as Equation (6.20), but the matrix C_{X_1} is a level k block circulant with circulant base blocks of size $(p \times p)$. Concretely, while considering all coefficients of B as elements of $(\mathbb{F}_p)^k$, we may write

$$\begin{aligned} B_p &= circ([\mathbf{0}, \mathbf{1}, \ldots \mathbf{p-1}]) \\ B_p^{\circ 2} &= circ(B_p, \mathbf{p} + B_p, \ldots (\mathbf{p-1})\mathbf{p} + B_p) \\ B_p^{\circ k} &= circ(B_p^{\circ k-1}, \mathbf{p^{k-1}} + B_p^{\circ k-1}, \ldots (\mathbf{p-1})\mathbf{p^{k-1}} + B_p^{\circ k-1}) \end{aligned}$$

with the *circ* function defined column-wise following [10], and $B_p^{\circ k}$ used as short-hand to indicate a matrix therein defined as belonging to the class $\mathcal{BCCB}(p, p, \ldots p)$, k times.

As shown in [10], matrices with this structure are diagonalised by $\mathbb{F}_p^{\otimes k}$. This naturally extends the known structure for binary random

variables, since as discussed in Section 6.2.1 the convolution matrix for variables in $(\mathbb{F}_2)^k$ is diagonalised by the Hadamard matrix H_{2^k}, which by construction is equal to $H_2^{\otimes k}$.

The Fourier matrix of size p can be written as a Vandermonde matrix of a primitive p-th root of unity as

$$F_p = \begin{bmatrix} \omega_p^{0\cdot 0} & \omega_p^{0\cdot 1} & \dots & \omega_p^{0\cdot(p-1)} \\ \omega_p^{1\cdot 0} & \omega_p^{1\cdot 1} & \ddots & \omega_p^{0\cdot 0} \\ \vdots & \ddots & \ddots & \vdots \\ \omega_p^{(p-1)\cdot 0} & \omega_p^{(p-1)\cdot 1} & \dots & \omega_p^{(p-1)^2} \end{bmatrix}$$

In other words, the entry in row r, column s is

$$F_p(r,s) = \omega_p^{rs}\,, \qquad r,s \in \mathbb{Z}_p\,.$$

By definition of the Kronecker product of two $(p \times p)$ matrices,

$$\begin{aligned} K &= M_1 \otimes M_2 \\ K(u,v) &= M_1(r_1,s_1)M_2(r_2,s_2) \quad u,v,r_i,s_i \in \mathbb{Z}_p \\ u &\equiv r_1 p + r_2 \qquad (6.23) \\ v &\equiv s_1 p + s_2 \qquad (6.24) \\ (F_p \otimes F_p)(u,v) &= \omega_p^{r_1 s_1}\omega_p^{r_2 s_2} \end{aligned}$$

which extends to the k-fold Kronecker product by induction using the p-ary representation of integers

$$F_p^{\otimes k}(u,v) = \omega_p^{\mathbf{r}\cdot\mathbf{s}}$$

for the specific $\mathbf{r},\mathbf{s}$ satisfying a polynomial in p such as (6.23) and (6.24) of degree $k-1$. In general, keeping either the row or column index fixed and iterating over the other means iterating over every element of $(\mathbb{F}_p)^k$; concretely, when evaluating the eigenvalues of a

probability mass $\mu \in \mathbb{R}^{p^k}$, the b-th eigenvalue corresponds to

$$\begin{aligned}\lambda(b) &= F_p^{\otimes k}(b,\cdot)\mu \\ &= \sum_{j=0}^{p^k-1} F_p^{\otimes k}(b,j)\mu(j) \\ &= \sum_{j=0}^{p^k-1} \omega_p^{\mathbf{b}\cdot\mathbf{j}}\mu(j)\,. \end{aligned} \tag{6.25}$$

Generalising from the case of the Walsh transform, this suggests the definition of the a-th order Fourier function evaluated at b as

$$f_a(b) = \omega_p^{\mathbf{b}\cdot\mathbf{a}}$$

(compare to Equation (6.7)), so that if $Y \in (\mathbb{F}_p)^k$ is the random variable with mass function μ, the eigenvalues may be written as

$$\lambda_Y(b) = \mathbb{E}\left[f_b(Y)\right]\,. \tag{6.26}$$

(compare to Equation (6.9)).

Lemma 149. *The probability mass function μ_Y of a random variable $Y \in (\mathbb{F}_p)^k$ satisfies*

$$\|\mu_Y - \mu_U\|_2 = \frac{1}{p^{k/2}}\|\lambda_Y - \lambda_U\|_2\,.$$

Proof.

$$\|\mu_Y - \mu_U\|_2 = \left\|\frac{F_p^{\otimes k}}{p^{k/2}}\left(\mu_Y - \mu_U\right)\right\|_2 \tag{6.27}$$

$$= \frac{1}{p^{k/2}}\left\|\lambda_Y - \lambda_U\right\|_2 \tag{6.28}$$

□

Corollary 150. *If the elements $Y(j) \in \mathbb{F}_p$ are independent but not necessarily identically distributed,*

$$\|\mu_Y - \mu_U\|_2 = \frac{1}{p^{k/2}}\left\|\bigotimes_{j=0}^{k-1}\lambda_{Y(j)} - \lambda_U\right\|_2$$

Proof. Since the $X(j)$ are independent, the probability mass function of $Y \in (\mathbb{F}_p)^k$ is

$$\begin{aligned}\mu_Y &= \mu_{Y(0)} \otimes \mu_{Y(1)} \otimes \cdots \mu_{Y(k-1)} \\ &= \bigotimes_{j=0}^{k-1} \mu_{Y(j)}\,,\end{aligned}$$

and the eigenvalues will be

$$\begin{aligned}\lambda_Y &= F_p^{\otimes n} \mu_Y \\ &= \bigotimes_{j=0}^{k-1} F_p \mu_{Y(j)} \\ &= \bigotimes_{j=0}^{k-1} \lambda_{Y(j)}\end{aligned}$$

where the second step follows by the mixed-product property of the Kronecker product.

□

We can now extend Theorem 139 as follows.

Theorem 151. *The total variation distance of a random vector $Y \in (\mathbb{F}_p)^k$ from uniform,*

$$\begin{aligned}\mathrm{TVD}(Y,\mathcal{U}) &= \frac{1}{2}\left\|\mu_Y - \mu_U\right\|_1 \\ &= \frac{1}{2}\delta_Y\end{aligned}$$

is bounded by

$$\delta_Y \leq \sum_{\mathbf{b}\in(\mathbb{F}_p)^k\setminus\mathbf{0}} \left|\prod_{u=0}^{k-1} \lambda_Y(\mathbf{b}(u))\right|\,. \tag{6.29}$$

Proof. Each eigenvalue may be written as

$$\lambda_Y(b) = \prod_{u=0}^{k-1} \lambda_Y(\mathbf{b}(u))\,.$$

The result follows directly from Lemma 149 and the known bound on vector spaces.

□

We can also extend Corollary 140 to establish a connection with the number of vectors of a specific Hamming weight, but in the non-binary case we can also go into more detail if the full composition of each vector in the space is known, as in the following definition.

Definition 152. Let $s(\mathbf{b})$ be the composition of $\mathbf{b} \in (\mathbb{F}_p)^k$ such that $s_j(\mathbf{b})$ is the number of components of $\mathbf{b}$ equal to j.

$$\begin{aligned} s(\mathbf{b}) &= (s_0, s_1, \dots s_{p-1}) \\ s_j &= |\{\, i \mid \mathbf{b}(i) = j \,\}| \end{aligned}$$

Let $W_{(\mathbb{F}_p)^k}(t)$ be the enumerator of the elements $\mathbf{b}$ having composition equal to t, with t being a p-tuple summing to k:

$$\begin{aligned} W_{(\mathbb{F}_p)^k}(t) &= |\,\{\, \mathbf{b} \in (\mathbb{F}_p)^k \mid s(\mathbf{b}) = t \,\}\,| \\ t &\in T \subset \mathbb{N}^p \\ \sum_j t_j &= k\,; \end{aligned}$$

then the number of $\mathbf{b}$ with Hamming weight equal to l is

$$A_l = \sum_t W(t) \qquad t \in \{\, t_0 = k - l \,\}\,.$$

In particular, if instead of $\mathbf{b} \in (\mathbb{F}_p)^k$ we consider a set of codewords $\mathbf{c} \in C$, the enumerator W_C is the complete weight enumerator of C, and A_l its weight distribution, as defined in [33] ch. 5§6.

Corollary 153. *If each $Y(j)$ is i.i.d., the total variation distance of a random vector $Y \in (\mathbb{F}_p)^k$ from uniform is bounded by*

$$\delta_Y \leq \sum_t W(t) \prod_{u=0}^{p-1} \left(\lambda_{Y(j)}(u)\right)^{t_u} \qquad t \in \{\, t_0 < k \,\} \tag{6.30}$$

Without knowledge of the full spectrum of $Y(j)$ one may obtain a coarser bound using the second largest eigenvalue is λ_{Y*}, as in Definition 145:

$$\delta_Y \leq \sum_{l=1}^{k} A_l \lambda_{Y*}^l \,, \tag{6.31}$$

where A_l is the number of $\mathbf{b} \in (\mathbb{F}_p)^k$ with Hamming weight $\mathrm{w}(\mathbf{b}) = l$.

Proof. Each eigenvalue may further be written as

$$\begin{aligned} \lambda_Y(b) &= \prod_{u=0}^{k-1} \lambda_Y(\mathbf{b}(u)) \\ &= \prod_{u=0}^{k-1} \sum_{v=0}^{p-1} \omega_p^{\mathbf{b}(u)\cdot v} \mu_{Y(u)}(v) \,, \end{aligned}$$

so all the $\mathbf{b}$ with identical composition t will correspond to equal eigenvalues, leading directly to Equation (6.30). If the Hamming weight $\mathrm{w}(\mathbf{b}(u)) = 0$, then the u-th term of the product will be equal to 1; Equation (6.31) follows by considering the worst case $\lambda_Y(j) = \lambda_{Y*}\ \forall j > 0$.

□

6.2.4 Fourier bound on entropy extractors

In order to arrive at a bound involving the distribution of weights, we begin by showing there is an unique association between code words and eigenvalues, just as there was with the bias of individual bits in the binary case (see Theorem 141).

If $Y = GX$, with X a random vector in $(\mathbb{F}_p)^n$, G a generator matrix of an (n, k, d) code over $\mathbb{F}_p$, we can establish a direct link

between eigenvalues of Y and codewords of G using Equation (6.26):

$$\begin{aligned}\lambda_Y(b) &= \mathbb{E}\left[f_b(GX)\right] \\ &= \sum_{j=0}^{p^n-1} \omega_p^{\mathbf{b}^T G\mathbf{j}} \mu_X(j) \\ &= \sum_{j=0}^{p^n-1} \omega_p^{\mathbf{c}\cdot\mathbf{j}} \mu_X(j) \end{aligned} \tag{6.32}$$

with $\mathbf{c} = \mathbf{b}^T G$ a particular word of the code. Note that choosing a particular $(k \times n)$ matrix G is equivalent to selecting the specific p^k rows specified by all the k codewords $\mathbf{c}$ that forms a subset of the p^n rows of $F_p^{\otimes n}$ by which to multiply μ_X.

Having noted this fundamental link in principle in the same manner as for the binary case (see Equation (6.16)), and having developed the required tools in Section 6.2.3, we can immediately state some more specific results for particular cases of practical interest, beginning with an extension of Theorem 151.

Theorem 154. *Let $Y = GX$, where $X \in (\mathbb{F}_p)^n$ is a random vector of length n, with each entry being an independent but not necessarily identically distributed variable $X(j) \in \mathbb{F}_p$ with mass function $\mu_{X(j)} \in \mathbb{R}^p$, and G is the generator matrix of an (n, k, d) linear code over $\mathbb{F}_p$. Then the b-th eigenvalue of the distribution of Y is*

$$\lambda_Y(b) = \prod_{j=0}^{n-1} \lambda_{X(j)}(\mathbf{c}(j)) \tag{6.33}$$

where $\mathbf{c}(j) \in \mathbb{F}_p$ is the j-th symbol in the codeword $\mathbf{c}^T = \mathbf{b}^T G$.

Proof. The specific combination corresponding to a word $\mathbf{c}$ is from

Equation (6.32):

$$\begin{aligned}\lambda_Y(b) &= \sum_{j=0}^{p^n-1} \omega_p^{\mathbf{c}\cdot\mathbf{j}} \mu_{X(j)} \\ &= \prod_{u=0}^{n-1} \sum_{v=0}^{p-1} \omega_p^{\mathbf{c}(u)\cdot\mathbf{v}} \mu_{X(u)}(v)\,.\end{aligned}$$

□

Corollary 155. *If all $X(j)$ are also i.i.d., the total variation distance of a random vector $Y \in (\mathbb{F}_p)^k$ from uniform is bounded by*

$$\delta_Y \leq \sum_t W_C(t) \prod_{u=0}^{p-1} \left(\lambda_{X(j)}(u)\right)^{t_u} \qquad t \in \{\, t_0 < n \,\} \tag{6.34}$$

Without knowledge of the full spectrum of $X(j)$ one may obtain a coarser bound using the second largest eigenvalue is λ_{X*}, as in Definition 145:

$$\delta_Y \leq \sum_{l=d}^{n} A_l \lambda_{X*}^l\,, \tag{6.35}$$

Here W_C and A_l are the complete weight enumerator and weight distribution of C, respectively, as in Definition 152.

Proof. The proof follows in the same manner as for Corollary 153.

□

The above can be viewed as a statement regarding the sum of n random variables, each in $\mathbb{F}_p$: if only $\mathrm{w}(\mathbf{c})$ symbols are nonzero, this corresponds to a sum of $\mathrm{w}(\mathbf{c})$ terms.

Corollary 156. *Using the minimum distance d, one may obtain the bound*

$$\delta_Y \leq (p^k - 1)\lambda_{X*}^d\,.$$

Note that all the results in this section extend to random vectors $X \in (\mathbb{F}_{p^m})^n$, that is to sequences of random vectors taking values in $\mathbb{F}_{p^m}$ by using the right matrix to diagonalise the convolution matrix of the sum of two such variables in order to compute the eigenvalues, and assuming the symbols of the generator matrix are taken in the same field, i.e. the code is chosen over $\mathbb{F}_{p^m}$. Following Section 6.2.3, this may be done using the Kronecker product $F_p^{\otimes mn}$.

Comparing Corollaries 143 and 156, it appears that a bound based solely on the minimum distance quickly risks becoming far from sharp as the dimension of the underlying random variable $X(j)$ increases.

6.2.5 Non-linear codes

As shown in [28], it is possible to construct ad-hoc non-linear maps with better properties than linear ones for specific cases; it was also noted that for a given compression ratio k/n of the output, there may exist non-linear codes with a greater minimum distance than any linear code. Since non-linear codes do not have a generator matrix G they are not straightforward to cover using the tools developed thus far, but we may use some of them to frame the fundamental issue with non-linear maps, as we see it, in terms of examining the distribution of the product of random variables. Consider the special case $X_1, X_0 \in \mathbb{F}_2$, and let their product be P_2; its mass function may be written as follows:

$$\begin{aligned} P_2 &= X_1 X_0 \\ \mu_{P_2} &= \begin{bmatrix} 1 & \mu_{X_1}(0) \\ 0 & \mu_{X_1}(1) \end{bmatrix} \begin{bmatrix} \mu_{X_0}(0) \\ \mu_{X_0}(1) \end{bmatrix} \end{aligned}$$

As long as neither X_0 or X_1 follow the categorical distribution with $\mathbb{P}(1) = 1$, the probability of their product being zero is strictly greater

than either of the initial probabilities. By induction,

$$\lim_{j\to\infty} \mu_{P_j} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

We may conclude that, while increasing the number of linear operations will lead to the uniform distribution in expectation, increasing the number of non-linear operations will in general lead to a worsening of the output distribution, except in very specific cases. By way of example, consider the two Bernoulli variables

$$B_+ \sim \mathcal{B}(2^{-1/2}), \qquad B_- \sim \mathcal{B}(1-2^{-1/2})$$
$$\mu_{B_+} = \begin{bmatrix} 1-2^{-1/2} \\ 2^{-1/2} \end{bmatrix} \qquad \mu_{B_-} = \begin{bmatrix} 2^{-1/2} \\ 1-2^{-1/2} \end{bmatrix}.$$

Note that the bias of these two random variables is identical; however, the distributions of their products are quite different:

$$\mu_{B_+B_+} = \begin{bmatrix} 2^{-1} \\ 2^{-1} \end{bmatrix} \qquad \mu_{B_-B_-} = \begin{bmatrix} 2^{1/2} - \frac{1}{2} \\ \frac{3}{2} - 2^{1/2} \end{bmatrix}.$$

While it is possible to find non-linear maps that are optimal in some specific cases, we observe that not only does repeated processing by non-linear maps in general lead to a worsening of the output, but it is also necessary to know or assume a specific distribution of the sequence to be processed to even attempt to find such a processing; even under the assumption of i.i.d. binary variables, knowledge of the bias of each bit is not sufficient.

Chapter 7

Other results

> *All truths are easy to understand once they are discovered; the point is to discover them.*
>
> Galileo Galilei

7.1 An improvement of known bounds on code parameters

Let C be an $(n, M, d)_q$ optimal code, namely there is no code with same distance and size as C whose length is strictly less than n. As seen in Chapter 2, many known bounds returns upper bounds on the size of codes once n and d are decided. Implicitly they refer to a code as an optimal one if it has the maximum number of codewords allowed to a code with chosen length and distance. We show how in some cases it is possible to improve many bounds by using our first definition of optimality.

Definition 157. Let k and d be positive integers. We denote with $n_q(k, d)$ a lower bound on the length $N_q(k, d)$ of an optimal code with combinatorial dimension k and minimum distance d.

Lemma 158. $N_q(k, d) \geq N_q(k-1, d) + 1$.

Proof. By contradiction, let $N_q(k, d) = N_q(k-1, d)$ and let $\bar{C}$ be an optimal $(N_q(k, d), M, d)_q$ code with combinatorial dimension k. We consider its subcodes $\bar{C}_\alpha$ of $\bar{C}$, with alpha ranging among all the elements of $\mathbb{F}_q$, consisting respectively of the codewords whose last coordinate is equal to α. One of the q subcodes has combinatorial dimension at least equal to $k-1$. This subcode is an $(N_q(k+1, d), M', d)_q$ code, $q^{k-2} < M'$, in which all codewords assume the same value in the last coordinate and can therefore be punctured. The result is an $(N_q(k+1, d) - 1, M')_q$ code whose distance is at least d. By Lemma 24 we can decrease its distance till we reach d and obtain an $(N_q(k+1, d) - 1, M', d)_2$ which contradicts the assumption $N_2(k+1, d) = N_2(k, d)$. □

Lemma 158 is a simple yet useful result, allowing improvements of known bounds.

Corollary 159. *For any d, $N_q(k+\Delta, d) \geq N_q(k,d)+\Delta$. Therefore a lower bound contradicting $n_q(k+\Delta,d) \geq n_q(k,d)+\Delta$ can be improved.*

We will make use of this corollary to obtain new results on the length of optimal codes by applying it to the following version of the Plotkin bound.

Theorem 160 (Plotkin Bound)**.**

$$n \geq \left\lceil \frac{qd}{q-1}\left(1-\frac{1}{M}\right)\right\rceil.$$

If a code has combinatorial dimension k, it has size at least equal to $q^{k-1}+1$, hence

$$n \geq \left\lceil \frac{qd}{q-1}\left(1-\frac{1}{q^{k-1}+1}\right)\right\rceil.$$

The Plotkin bound, in its original formulation, can only be applied to codes whose lengths are smaller than $\frac{qd}{q-1}$. As a consequence of this requirement, the bound in Theorem 160 becomes useless whenever $\frac{qd}{q-1} < q^{k-1}+1$, i.e. whenever $k > \log_q d + 1$. In this case Theorem 160 always return the value $\frac{qd}{q-1}$. This however contradicts Lemma 158 and leads us to the following result.

Theorem 161 (Bound D)**.** *Let $\omega = \frac{q}{q-1}$, let $k > \log_q d + 1$ and let $r = \lfloor \log_q d \rfloor + 1$. Then*

$$n \geq \omega d + k - r - 1.$$

Proof. It follows by applying Lemma 158 to the Plotkin bound in Theorem 160. □

Notice that the same arguments can be applied to codes whose combinatorial dimension is k, even though $M < q^k$. We obtain the following corollary.

Corollary 162. *Let $n \geq \omega d$. Then*

$$A_q(n,d) \leq q^{n-\omega d+r+2} - 1,$$

where $A_q(n,d)$ is the maximum number of codewords of a code with length n and minimum distance d, while $r = \lfloor \log_q d \rfloor + 1$.

Proof. Let C be an optimal $(n, A_q(n,d), d)_q$ code, with $n \geq \omega d$, and let k be the combinatorial dimension of C. If $q^k > \omega d$ then $k > \log_q d + 1$, the bound follows from Theorem 161. Otherwise we have $q^k \leq \omega d$, and we observe that

$$\omega d < q^{r+1} - 1 \leq q^{n-\omega d+r+2} - 1,$$

implying that $k \leq n - \omega d + r + 1$. □

In Table 7.1 we list several values of n and d for which the bound for $A_2(n,d)$ in Corollary 162 outperforms some known bounds in the case of binary codes. The comparison is made between the Elias bound, the Johnson bound and the Hamming bound. The list of parameters is not exhaustive, as the list of bounds for which the comparison can be done. From the brief experimental analysis, Corollary 162 provides stronger upper bounds for $A_2(n,d)$ than other bounds in a small range of lengths close to twice the minimum distance d. More detailed tables can be found in Appendix A.

7.2 The binary Möbius transform

Let $f : (\mathbb{F}_2)^k \to \mathbb{F}_2$ be a Boolean function. It is known that f can be represented in an unique way as a square-free polynomial in $\mathbb{F}_2[x_1, \ldots, x_k]$, and so it is uniquely determined by a set of 2^k binary coefficients.

Let us consider now the module $R = \mathbb{F}_2[A]$, where A is a set of variables a_e whose indices are elements $e \in (\mathbb{F}_2)^k$, and let f be a function $(\mathbb{F}_2)^k \to R$. As above, f is uniquely identified as a square-free

n	d	EB	JB	HB	Cor. 162
21	10	158	162	277	127
22	10	299	257	460	255
25	12	198	293	490	127
26	12	344	460	801	255
29	14	229	523	863	127
30	14	425	818	1397	255
37	18	320	1642	2656	255
38	18	551	2550	4237	511
41	20	349	2890	4641	255
42	20	633	4476	7365	511
45	22	385	5069	8094	255
46	22	719	7832	12786	511
49	24	428	8865	14087	255
50	24	770	13672	22169	511

Table 7.1: Some values of (d, n) for which Corollary 162 outperforms some known bounds for binary codes. EB stands for Elias bound, JB for Johnson bound and HB for Hamming bound.

polynomial in $R[x_1, \ldots, x_k]$, and we are interested in the coefficients of f.
Let us denote with $X = (x_1, \ldots, x_k)$, $e = (e_1, \ldots, e_k) \in (\mathbb{F}_2)^k$ and $X^e = x_1^{e_1} \cdots x_k^{e_k}$. We can therefore describe f as the polinomial

$$f(X) = \sum_{e \in (\mathbb{F}_2)^k} a_e X^e, \tag{7.1}$$

where $a_e \in R$.
We are interested in the map

$$(\mathbb{F}_2)^k \ni e \mapsto a_e \in R.$$

This map can itself be described as a square-free polynomial function

$$\mathrm{a}_f : (\mathbb{F}_2)^k \to R.$$

Definition 163. Let $f \in R[X]$ be a square-free polynomial. We denote with $\mathrm{a}_f \in R[e]$ the square-free polynomial associated to the function $(\mathbb{F}_2)^k \to R$ that send each $e \in (\mathbb{F}_2)^k$ to the coefficient a_e corresponding to the monomial $a_e X^e$ in f. The function a_f is called the binary Möbius transform of f.

The classical description of the binary Möbius transform arises from observing Equation 7.1. The value that f assumes on a given point $b \in (\mathbb{F}_2)^n$ can be computed as

$$f(b) = \sum_{e \preccurlyeq b} a_e,$$

where $e \preccurlyeq b$ if $\mathrm{supp}(e) \subseteq \mathrm{supp}(b)$. This follows directly from the fact that the evaluation at b of a given monomial X^e in f is non-zero if and only if all variables in X^e are non-zero at b, hence if and only if $\mathrm{supp}(e) \preccurlyeq \mathrm{supp}(b)$. Interesting enough, the converse is also true, namely

$$a_e = \sum_{b \preccurlyeq e} f(b).$$

The proof of this fact can found e.g. in [9], alongside a divide-and-conquer butterfly algorithm to compute the binary Möbius function of a given f. We remark that the algorithm is described as a way to compute the Absolute Normal Form of a Boolean function from its truth table.

We present here a closed formula to obtain the absolute normal form of the binary Möbius transform of a function. To our knowledge this formula does not appear in previous published works.

Theorem 164. *The binary Möbius transform is a map from $R[X]$ to itself, which can be defined by the following map:*

$$\begin{array}{rccc} \mathrm{a}: & \mathbb{F}_2[X] & \to & \mathbb{F}_2[X] \\ & f & \mapsto & \mathrm{a}_f = (1+X)\cdot f\left(\frac{X}{1+X}\right) \end{array} \tag{7.2}$$

Proof. Using the notation in Equation 7.1, $\mathrm{a}_f(e) = a_e$, where a_e is the coefficient in f of the monomial X^e. By Formula (7.2) the monomial X^e is mapped to the polynomial $X^e(1+X)^{1+e}$, which is the only polynomial equal to 1 on $e \in (\mathbb{F}_2)^n$ only. □

Example 4. We consider the Boolean function $(\mathbb{F}_2)^2 \to (\mathbb{F}_2)^1$ defined by $f(x_1, x_2) = 1 + x_1 + x_1x_2$. From a_f we expect $\mathrm{a}_f(0,0) = 1$, $\mathrm{a}_f(1,0) = 1$, $\mathrm{a}_f(0,1) = 0$ and $\mathrm{a}_f(1,1) = 1$. By Formula 7.2 we obtain

$$\begin{aligned} a_f &= (1+x_1)(1+x_2)\, f\left(\tfrac{x_1}{1+x_1}, \tfrac{x_2}{1+x_2}\right) \\ &= (1+x_1)(1+x_2)\left(1 + \tfrac{x_1}{1+x_1} + \tfrac{x_1x_2}{(1+x_1)(1+x_2)}\right) \\ &= (1+x_1)(1+x_2) + x_1(1+x_2) + x_1x_2 \\ &= 1 + x_2 + x_1x_2. \end{aligned}$$

The truth table of the obtained function is as expected.

Using the definition of a_f, the function f can be written as

$$f(X) = \sum_{e \in (\mathbb{F}_2)^k} \mathrm{a}_f(e) X^e.$$

Proposition 165. a *is an involution.*

Proof. By Theorem 164 we have

$$\mathrm{a}_f(x_1 \ldots, x_k) = (1+x_1)\cdots(1+x_k)\cdot f\left(\frac{x_1}{1+x_1}, \ldots, \frac{x_k}{1+x_k}\right),$$

then,

$$\mathrm{a}(\mathrm{a}_f) = \prod_{j=1}^{k}(1+x_j)\cdot\prod_{h=1}^{k}\left(1+\frac{x_h}{1+x_h}\right)\cdot f\left(\frac{\frac{x_1}{1+x_1}}{1+\frac{x_1}{1+x_1}}, \ldots, \frac{\frac{x_k}{1+x_k}}{1+\frac{x_k}{1+x_k}}\right).$$

Since $1+\frac{x}{1+x} = \frac{1}{1+x}$, by computation we obtain

$$\mathrm{a}(\mathrm{a_f})(x_1, \ldots, x_k) = f(x_1, \ldots, x_k).$$

□

Suppose now to know the evaluation vector v of a map

$$f : (\mathbb{F}_2)^k \to \mathbb{F}_2\,.$$

Due to Proposition 165, we known that v is the set of the coefficients of the binary Möbius transform of f, so

$$\mathrm{a}_f = \sum_{e\in(\mathbb{F}_2)^k} v_e X^e.$$

By applying the binary Möbius transform to a_f we can retrieve f.

Example 5. Let f be a polynomial map from $(\mathbb{F}_2)^3$ to $\mathbb{F}_2$, defined by the following relationships:

$$\begin{cases} f(0,0,0) = 1 \\ f(0,0,1) = 0 \\ f(0,1,0) = 1 \\ f(0,1,1) = 0 \\ f(1,0,0) = 0 \\ f(1,0,1) = 1 \\ f(1,1,0) = 0 \\ f(1,1,1) = 0 \end{cases}$$

Denoting with $X = (x_1, x_2, x_3)$, the binary Möbius transform of f is therefore

$$\mathrm{a}_f = X^{(0,0,0)} + X^{(0,1,0)} + X^{(1,0,1)} = 1 + x_2 + x_1x_3.$$

We apply the transform to $1 + x_2 + x_1x_3$, namely

$$f = (1 + x_1)(1 + x_2)(1 + x_3)\left(1 + \frac{x_2}{1 + x_2} + \frac{x_1x_3}{(1 + x_2)(1 + x_3)}\right),$$

hence

$$f = 1 + x_1 + x_3 + x_1x_2x_3.$$

We consider now the following two problems:

Problem 1:

Let $L = \{f_1, \ldots, f_m\}$ be polynomials in $\mathbb{F}_2[X]$, with $X = (x_1, \ldots, x_n)$. We consider the ideal I generated by L and we want to prove that the variety of I contains points in $(\mathbb{F}_2)^n$. This is equivalent to show that the variety of the ideal $J = \langle L, E\rangle$ is not empty, where $E = \{x_1^2 + x_1, \ldots, x_n^2 + x_n\}$ is the set of the field equations.

Problem 2:

Let $F = (f_1(X), \ldots, f_m(X))$ be a vectorial Boolean map $(\mathbb{F}_2)^n \to (\mathbb{F}_2)^m$. We want to show that the zero vector belongs to the image of F.

Proposition 166. *Problem 1 and Problem 2 are equivalent.*

Proof. Let P be a point of the variety of J. Since $E \subset J$, P belongs to $(\mathbb{F}_2)^n$ and since $L \subset J$ then $f_i(P) = 0$ for each $i = 1, \ldots, m$. This implies that the Boolean function $F = (f_1, \ldots, f_m)$ evaluated at P is the zero vector.

On the other hand, each point Q sent to zero by F are elements of $(\mathbb{F}_2)^n$ for which $f_i(Q) = 0$ for all $i = 1, \ldots, m$. This implies that they belong to the variety of J. □

Lemma 167. *Let g and h be two Boolean functions and let v_g and v_h be their evaluation vectors. The evaluation vector v_ϕ of the map $\phi = 1 + (1 + g)(1 + h)$ is given by the OR between v_g and v_h.*

Proof. $\phi(P) = 0$ if and only if both f and g are zero at P. This is equivalent to say that both v_g and v_h have a zero in the same coordinate j if and only if v_ϕ is zero in the coordinate j. □

Theorem 168. *Let $F = (f_1, \ldots, f_m)$ be a vectorial Boolean function. Then $0 \in \mathrm{Im}(F)$ if and only if the Boolean function*

$$\bar{\varphi} = 1 + (1 + f_1) \cdots (1 + f_m)$$

is not the constant map 1.

Proof. $0 \in \mathrm{Im}(F)$ if and only if the evaluation vectors $v_1, \ldots, v_m$ of the Boolean functions $f_1, \ldots, f_m$ have at least a zero in a common position. By Lemma 167 this can happen if and only if the evaluation vector of $\bar{\varphi}$ have at least a zero, and the only map without zeros in its evaluation vector is the constant map 1. □

Corollary 169. *Let $L = \{f_1, \ldots, f_m\}$ be polynomials in $\mathbb{F}_2[X]$, let $E = \{x_1^2 + x_1, \ldots, x_n^2 + x_n\}$ be the set of the field equations and let $J = \langle L, E \rangle$. Then $J = \langle 1 \rangle$ if and only the map $\varphi = (1 + f_1) \cdots (1 + f_m)$ is equal to* 0.

Proof. It follows directly from Proposition 166 and Theorem 168. □

We remark that Corollary 169 provides a method for testing whether the variety of an ideal J in $\mathbb{F}_2[X]$ containing the field equations is empty or not, hence it is equivalent to Hilbert's Nullstellensatz.

7.3 A probabilistic algorithm for the weight distribution

We considered in Chapter 6.2 the use of linear code generator matrices as whitening functions for the output of a random number generator, and showed therein that the weight distribution of the code uniquely determines the distribution of the resulting random variable.

It is unfortunate that weight distributions are not readily available for many codes; we therefore turn the problem on its head and use random output of a specified quality to estimate the weight distribution. We note from the outset that this is a highly inefficient solution to the problem of computing weight distributions, costing a few orders of magnitude more than simply enumerating every single weight; we describe it purely for the sake of curiosity, and in the hope that improvements may be found. Furthermore, although the framework can be extended to non-binary codes, the inefficiency of the algorithm prompts us to consider the binary case only for the moment.

We begin by recalling how we can tie the distribution of the input random bit stream X to that of the processed output $Y = GX$, with G the generator matrix of a linear binary $[n, k, d]_2$ code used as a compression function on n bits of X. The characteristic function of a random variable with mass function μ is its inverse Fourier transform, which in the binary case is the Hadamard or Walsh transform:

$$\chi_Y = H\mu_Y$$

Let the balance of a random bit be defined as $\beta = 1 - 2\mathbb{E}[B]$. This definition is not usual in the sense that it is not symmetric, choosing the opposite order of bits will change the sign; the more usual quantity to measure is the bias $\varepsilon/2 = |\beta|/2$, but we here require β as above.

We can summarise our main reasoning as follows:

1. the b-th row of $H\mu_Y$ corresponds to the c-th row of $H\mu_X$, selected by the codeword $c^T = b^T G$;

2. if the individual bits of X are i.i.d. with balance β, the c-th row of $H\mu_X$ equals the characteristic function of a linear combination of $\mathrm{w}(c)$ bits, with $\mathrm{w}(c)$ the Hamming weight of c; hence, the b-th row of $H\mu_Y$ is equal to $\beta^{\mathrm{w}(c)}$.

3. If we don't know the full weight distribution $\{A_l\}_l$ of the corresponding code $\mathcal{C}$ but we do know the generator matrix G, we

can apply the compression function to a stream of independent bits of specified balance and estimate A from the characteristic function of the compressed stream Y.

Item 1 can be seen as follows. The b-th row of the transform can be written as an expected value in terms of the b-th Walsh function

$$\begin{aligned}
\chi_Y(b) &= \mathbb{E}[h_b(Y)] \\
&= \mathbb{E}[(-1)^{\mathbf{b}\cdot Y}] \\
&= \mathbb{E}[(-1)^{\mathbf{b}\cdot GX}] \\
&= \sum_{x=0}^{2^n-1} (-1)^{\mathbf{b}\cdot G\mathbf{x}} \mu_X(x) \\
&= \sum_{x=0}^{2^n-1} (-1)^{\mathbf{c}\cdot\mathbf{x}} \mu_X(x) \\
&= \mathbb{E}[h_c(X)]
\end{aligned}$$

Item 2 relies on the assumption of independence. The Hadamard matrix of size 2^k can always be decomposed using the Kronecker product; the same is true for the mass function μ_X only if each X_j is independent of the others:

$$\begin{aligned}
H_{2^k} &= H_2^{\otimes k} \\
\mu_X &= \mu_{X_j}^{\otimes k} \\
H_{2^k}\mu_X &= \bigotimes_{j=0}^{n} H_2\mu_{X_j}
\end{aligned}$$

If each X_j is also identically distributed, then each $H_2\mu_{X_j} = (1, \beta)^T$. By the ordering induced by the Kronecker product, considering row c of the result, the value 1 will be selected in the j-th term of the product whenever the j-th bit of the binary representation of c is equal to 0, whence we deduce that $\chi_X(c) = \beta^{\mathrm{w}(c)}$.

Note that the zeroth word will always lead to a value of 1, which we can safely ignore by considering the variable $\chi = \chi_Y - \chi_{\mathcal{U}}$, and

the uniform distribution over whatever space Y is defined will always have a bias of 0. Assuming no special ordering of the words can be found, we can treat χ as a random variable to be sampled.
The resulting algorithm is as follows:

1. generate N strings x of n independent random bits each with a fixed balance β;
2. compute $y = Gx$ for each string;
3. estimate the mass function of Y using the N resulting samples;
4. compute $\hat{l} = \log_{\varepsilon} |H_2^{\otimes k} \mu_Y|$ to recover the exponents estimating the weights;
5. round $\hat{l}$ to the nearest integer.

The above is of course specific to binary codes, for instance in the use of β and the absolute value.

This algorithm clearly compares unfavourably to the deterministic, brute-force solution: given G, one can simply compute every single word of the code by cycling through all 2^k possible messages y and performing a matrix multiplication $y^T G$ for each of them. This yields the full list of codewords, from which A may be immediately deduced. By contrast, as described above we need to estimate the probability of each y occurring by repeated sampling, which will likely take 100 samples per message, depending on the required accuracy, meaning $100 \cdot 2^k$ multiplications Gx. We can see an example of this in Figures 7.1 and 7.2, showing χ_Y after applying the generator matrix of a BCH(7,4,3) code as a compression function on bitstreams with a range of β and two separate values of N. For reference, the weight distribution of this code is $A = [1, 0, 0, 7, 7, 0, 0, 1]$. To provide a more realistic example we also considered the generator matrix of the BCH(33, 13, 5) code with increasing sample size; results are exemplified in Figures 7.3 and 7.4. The estimated distribution does not appear to be close

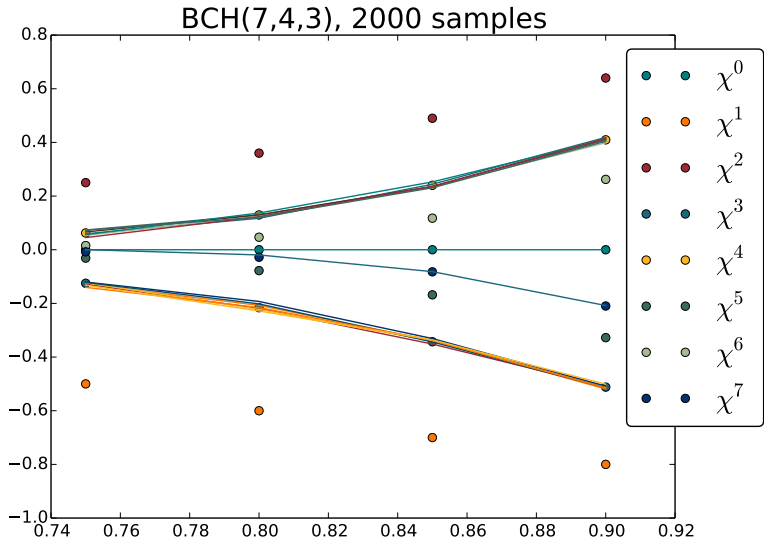

Figure 7.1: Row values of the characteristic function of the processed bit stream, as a function of the $\mathbb{P}(1)$ of each of the i.i.d. bits of the stream, compared with powers of χ_j. These results are obtained using 2000 samples.

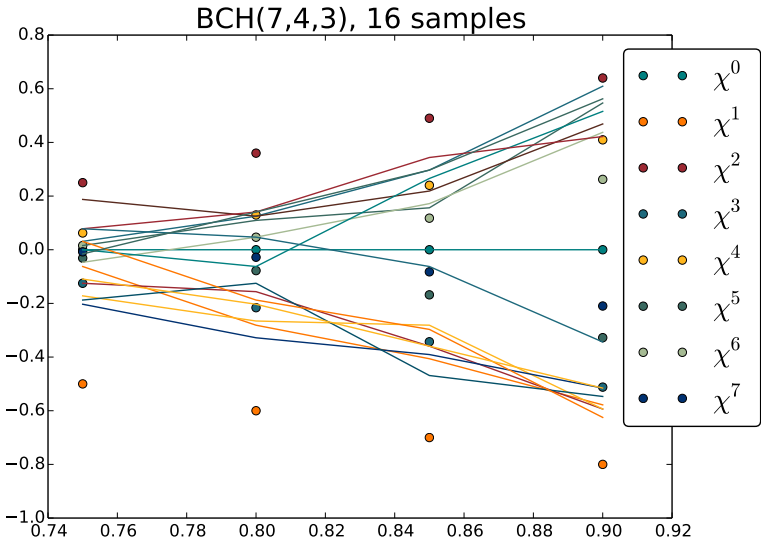

Figure 7.2: Row values of the characteristic function of the processed bit stream, as a function of the $\mathbb{P}(1)$ of each of the i.i.d. bits of the stream, compared with powers of χ_j. These results are obtained using $2^k = 16$ samples.

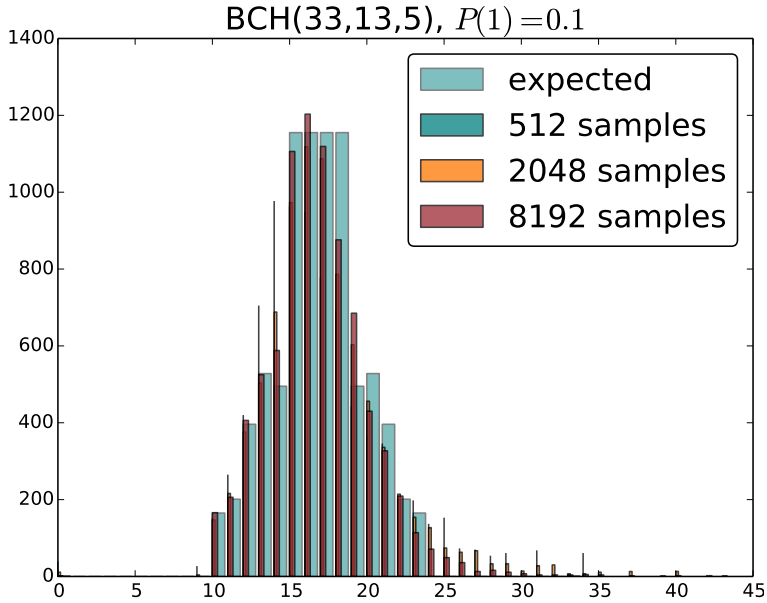

Figure 7.3: Application of the weight estimation algorithm by processing of a bit stream with $\mathbb{P}(1) = 0.1$ by the generator matrix of the BCH(33, 13, 5) code.

to the known values even with 2^{k+2} samples (recall that 2^k operations would be the brute-force deterministic method cost). The overestimate of high weights does not appear to be related to a high $\mathbb{P}(1)$, i.e. a higher likelihood of sampling high-weight codewords. We also note that even with 2^k samples the algorithm fails to correctly estimate the minimum distance, returning a non-zero estimate of words of a smaller weight.

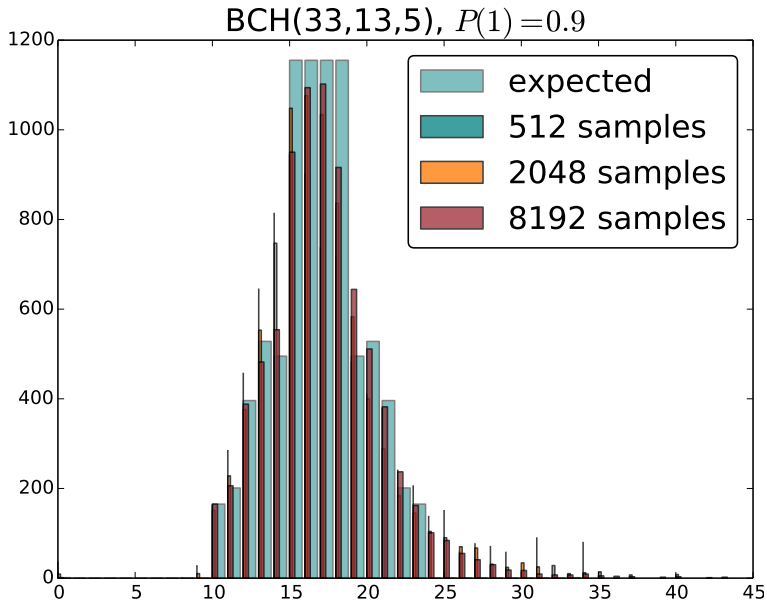

Figure 7.4: Application of the weight estimation algorithm by processing of a bit stream with $\mathbb{P}(1) = 0.9$ by the generator matrix of the BCH(33, 13, 5) code.

Appendix A

Tables of bounds

In this chapter we list some values obtained from some bounds presented in this work. We use the following notations:

EB Elias bound.

JB Johnson bound.

SB Singleton bound.

HB Hamming bound.

PB Plotkin bound.

bA bound A, as in Proposition 109.

bB bound B. We use Corollary 112 for the binary case in Section A.1, and Corollary 111 in the other sections.

bC bound C, as in Proposition 113.

bD bound D, as in Theorem 161.

Each section deals with codes respectively over $\mathbb{F}_2$, $\mathbb{F}_3$, $\mathbb{F}_4$, $\mathbb{F}_5$, $\mathbb{F}_7$ and $\mathbb{F}_8$. Each table contains lower bounds for a fixed distance and several combinatorial dimensions.
To ease the comparison, each bound is converted to a lower bound on $N_q(q^k, d)$, and in each table, for each dimension k, we highlight in bold the stronger lower bounds on n. Notice that bound A and bound

B are bounds for $S_q(k, d)$.

We remark that most of the distance-dimension pairs used in the following sections are outside the Plotkin range, and in such cases the Plotkin bound is constant.

A.1 Bounds for codes in $\mathbb{F}_2$

	$q = 2, \quad d = 4$								
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	**7**	**7**	6	6	**7**	**7**	**7**	3	-
4	**8**	**8**	7	7	**8**	**8**	**8**	3	**8**
5	9	**10**	8	9	8	9	9	3	9
6	10	**11**	9	10	8	10	10	3	10
7	11	**12**	10	11	8	11	11	3	11
8	12	**13**	11	12	8	12	12	3	12
9	13	**14**	12	13	8	13	13	3	13
10	14	**15**	13	14	8	14	14	3	14
11	**16**	**16**	14	15	8	15	15	3	15
12	17	**18**	15	17	8	16	16	3	16
13	18	**19**	16	18	8	17	17	3	17
14	19	**20**	17	19	8	18	18	3	18
15	20	**21**	18	20	8	19	19	3	19
16	21	**22**	19	21	8	20	20	7	20
17	22	**23**	20	22	8	21	21	7	21
18	23	**24**	21	23	8	22	22	7	22
19	24	**25**	22	24	8	23	23	7	23
20	25	**26**	23	25	8	24	24	7	24
21	26	**27**	24	26	8	25	25	7	25
22	27	**28**	25	27	8	26	26	7	26
23	28	**29**	26	28	8	27	27	7	27
24	29	**30**	27	29	8	28	28	7	28
25	30	**31**	28	30	8	29	29	7	29
26	**32**	**32**	29	31	8	30	30	7	30
27	33	**34**	30	33	8	31	31	7	31
28	34	**35**	31	34	8	32	32	7	32

Table A.1: Bounds for codes with $q = 2$, $d = 4$, $3 \leq k \leq 28$.

$q = 2, \quad d = 6$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	10	10	8	9	**11**	10	**11**	3	-
4	11	**12**	9	10	**12**	11	**12**	3	**12**
5	12	**13**	10	12	12	12	**13**	3	**13**
6	13	**14**	11	13	12	13	**14**	3	**14**
7	14	**15**	12	14	12	14	**15**	3	**15**
8	15	**16**	13	15	12	15	**16**	3	**16**
9	17	**18**	14	17	12	16	17	3	17
10	18	**19**	15	18	12	17	18	3	18
11	19	**20**	16	19	12	18	19	3	19
12	20	**21**	17	20	12	19	20	3	20
13	21	**22**	18	21	12	20	21	3	21
14	22	**24**	19	22	12	21	22	3	22
15	24	**25**	20	24	12	22	23	3	23
16	25	**26**	21	25	12	23	24	9	24
17	26	**27**	22	26	12	24	25	9	25
18	27	**28**	23	27	12	25	26	9	26
19	28	**29**	24	28	12	26	27	9	27
20	29	**30**	25	29	12	27	28	9	28
21	30	**31**	26	30	12	28	29	9	29
22	31	**33**	27	31	12	29	30	9	30
23	33	**34**	28	33	12	30	31	9	31
24	34	**35**	29	34	12	31	32	9	32
25	35	**36**	30	35	12	32	33	9	33
26	36	**37**	31	36	12	33	34	9	34
27	37	**38**	32	37	12	34	35	9	35
28	38	**39**	33	38	12	35	36	9	36
29	39	**40**	34	39	12	36	37	9	37
30	40	**41**	35	40	12	37	38	9	38

Table A.2: Bounds for codes with $q = 2$, $d = 6$, $3 \leq k \leq 30$.

$q = 2, \quad d = 8$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	13	12	10	11	**14**	13	**14**	6	-
4	**15**	14	11	13	**15**	14	**15**	7	-
5	**16**	**16**	12	14	**16**	15	**16**	7	**16**
6	16	**17**	13	16	16	16	**17**	7	**17**
7	**18**	**18**	14	17	16	17	**18**	7	**18**
8	19	**20**	15	18	16	18	19	7	19
9	20	**21**	16	20	16	19	20	7	20
10	21	**22**	17	21	16	20	21	7	21
11	22	**23**	18	22	16	21	22	7	22
12	23	**24**	19	23	16	22	23	7	23
13	25	**26**	20	25	16	23	24	7	24
14	26	**27**	21	26	16	24	25	7	25
15	27	**28**	22	27	16	25	26	7	26
16	28	**29**	23	28	16	26	27	15	27
17	29	**31**	24	29	16	27	28	15	28
18	31	**32**	25	31	16	28	29	15	29
19	32	**33**	26	32	16	29	30	15	30
20	33	**34**	27	33	16	30	31	15	31
21	34	**35**	28	34	16	31	32	15	32
22	35	**36**	29	35	16	32	33	15	33
23	36	**37**	30	36	16	33	34	15	34
24	37	**39**	31	38	16	34	35	15	35
25	39	**40**	32	39	16	35	36	15	36
26	40	**41**	33	40	16	36	37	15	37
27	41	**42**	34	41	16	37	38	15	38
28	42	**43**	35	42	16	38	39	15	39
29	43	**44**	36	43	16	39	40	15	40
30	44	**45**	37	44	16	40	41	15	41

Table A.3: Bounds for codes with $q = 2$, $d = 8$, $3 \leq k \leq 30$.

$q = 2, \quad d = 10$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	16	15	12	14	**18**	16	16	5	-
4	18	16	13	15	**19**	17	17	5	-
5	19	18	14	17	**20**	18	18	5	**20**
6	20	20	15	18	20	19	19	5	**21**
7	21	21	16	20	20	20	20	5	**22**
8	22	22	17	21	20	21	21	5	**23**
9	**24**	**24**	18	23	20	22	22	5	**24**
10	**25**	**25**	19	24	20	23	23	5	**25**
11	**26**	**26**	20	25	20	24	24	5	**26**
12	27	**28**	21	27	20	25	25	5	27
13	28	**29**	22	28	20	26	26	5	28
14	29	**30**	23	29	20	27	27	5	29
15	30	**32**	24	30	20	28	28	5	30
16	32	**33**	25	32	20	29	29	15	31
17	33	**34**	26	33	20	30	30	15	32
18	34	**35**	27	34	20	31	31	15	33
19	35	**36**	28	35	20	32	32	15	34
20	36	**38**	29	37	20	33	33	15	35
21	38	**39**	30	38	20	34	34	15	36
22	39	**40**	31	39	20	35	35	15	37
23	40	**41**	32	40	20	36	36	15	38
24	41	**42**	33	41	20	37	37	15	39
25	42	**44**	34	42	20	38	38	15	40
26	43	**45**	35	44	20	39	39	15	41
27	45	**46**	36	45	20	40	40	15	42
28	46	**47**	37	46	20	41	41	15	43
29	47	**48**	38	47	20	42	42	15	44
30	48	**49**	39	48	20	43	43	15	45

Table A.4: Bounds for codes with $q = 2$, $d = 10$, $3 \leq k \leq 30$.

$q = 2, \quad d = 12$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	19	18	14	16	**21**	19	**21**	9	-
4	22	19	15	18	**23**	20	**23**	9	-
5	23	21	16	20	**24**	21	**24**	9	**24**
6	24	22	17	21	24	22	**25**	9	**25**
7	25	24	18	23	24	23	**26**	9	**26**
8	26	25	19	24	24	24	**27**	9	**27**
9	27	27	20	26	24	25	**28**	9	**28**
10	28	28	21	27	24	26	**29**	9	**29**
11	29	**30**	22	28	24	27	**30**	9	**30**
12	**31**	**31**	23	30	24	28	**31**	9	**31**
13	**32**	**32**	24	31	24	29	**32**	9	**32**
14	33	**34**	25	32	24	30	33	9	33
15	34	**35**	26	34	24	31	34	9	34
16	**36**	**36**	27	35	24	32	35	21	35
17	**37**	**37**	28	36	24	33	36	21	36
18	38	**39**	29	37	24	34	37	21	37
19	39	**40**	30	39	24	35	38	21	38
20	40	**41**	31	40	24	36	39	21	39
21	41	**42**	32	41	24	37	40	21	40
22	42	**43**	33	42	24	38	41	21	41
23	43	**45**	34	44	24	39	42	21	42
24	45	**46**	35	45	24	40	43	21	43
25	46	**47**	36	46	24	41	44	21	44
26	47	**48**	37	47	24	42	45	21	45
27	48	**49**	38	48	24	43	46	21	46
28	49	**51**	39	50	24	44	47	21	47
29	51	**52**	40	51	24	45	48	21	48
30	52	**53**	41	52	24	46	49	21	49

Table A.5: Bounds for codes with $q = 2$, $d = 12$, $3 \leq k \leq 30$.

$q = 2, \quad d = 14$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	23	20	16	18	**25**	22	**25**	7	-
4	25	21	17	20	**27**	23	**27**	7	-
5	27	23	18	22	**28**	24	**28**	7	**28**
6	28	25	19	24	28	25	**29**	7	**29**
7	29	27	20	25	28	26	**30**	7	**30**
8	30	28	21	27	28	27	**31**	7	**31**
9	31	29	22	28	28	28	**32**	7	**32**
10	32	31	23	30	28	29	**33**	7	**33**
11	33	32	24	31	28	30	**34**	7	**34**
12	**35**	34	25	33	28	31	**35**	7	**35**
13	**36**	35	26	34	28	32	**36**	7	**36**
14	**37**	36	27	35	28	33	**37**	7	**37**
15	**38**	**38**	28	37	28	34	**38**	7	**38**
16	**39**	**39**	29	38	28	35	**39**	21	**39**
17	**41**	**41**	30	39	28	36	40	21	40
18	**42**	**42**	31	41	28	37	41	21	41
19	**43**	**43**	32	42	28	38	42	21	42
20	**44**	**44**	33	43	28	39	43	21	43
21	45	**46**	34	44	28	40	44	21	44
22	**47**	**47**	35	46	28	41	45	21	45
23	**48**	**48**	36	47	28	42	46	21	46
24	**49**	**49**	37	48	28	43	47	21	47
25	**50**	**50**	38	49	28	44	48	21	48
26	51	**52**	39	51	28	45	49	21	49
27	52	**53**	40	52	28	46	50	21	50
28	53	**54**	41	53	28	47	51	21	51
29	54	**55**	42	54	28	48	52	21	52
30	**56**	**56**	43	55	28	49	53	21	53

Table A.6: Bounds for codes with $q = 2$, $d = 14$, $3 \leq k \leq 30$.

$q = 2, \quad d = 16$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	25	22	18	21	**28**	25	**28**	12	-
4	29	24	19	23	**30**	26	**30**	14	-
5	**31**	26	20	25	**31**	27	**31**	15	-
6	**32**	28	21	26	**32**	28	**32**	15	**32**
7	32	29	22	28	32	29	**33**	15	**33**
8	33	31	23	30	32	30	**34**	15	**34**
9	**35**	32	24	31	32	31	**35**	15	**35**
10	**36**	34	25	33	32	32	**36**	15	**36**
11	**37**	35	26	34	32	33	**37**	15	**37**
12	**38**	37	27	36	32	34	**38**	15	**38**
13	**39**	38	28	37	32	35	**39**	15	**39**
14	**41**	40	29	38	32	36	40	15	40
15	**42**	41	30	40	32	37	41	15	41
16	**43**	42	31	41	32	38	42	31	42
17	**44**	**44**	32	42	32	39	43	31	43
18	**45**	**45**	33	44	32	40	44	31	44
19	**47**	46	34	45	32	41	45	31	45
20	**48**	**48**	35	46	32	42	46	31	46
21	**49**	**49**	36	48	32	43	47	31	47
22	**50**	**50**	37	49	32	44	48	31	48
23	**52**	51	38	50	32	45	49	31	49
24	**53**	**53**	39	52	32	46	50	31	50
25	**54**	**54**	40	53	32	47	51	31	51
26	**55**	**55**	41	54	32	48	52	31	52
27	**56**	**56**	42	55	32	49	53	31	53
28	**58**	**58**	43	57	32	50	54	31	54
29	**59**	**59**	44	58	32	51	55	31	55
30	**60**	**60**	45	59	32	52	56	31	56

Table A.7: Bounds for codes with $q = 2$, $d = 16$, $3 \leq k \leq 30$.

$q = 2, \quad d = 18$

k	EB	JB	SB	HB	PB	bA	bB	bC	bD
3	29	24	20	23	**32**	28	30	9	-
4	33	27	21	25	**34**	29	32	9	-
5	**35**	28	22	27	**35**	30	33	9	-
6	35	30	23	29	**36**	31	34	9	**36**
7	36	32	24	31	36	32	35	9	**37**
8	37	34	25	32	36	33	36	9	**38**
9	38	35	26	34	36	34	37	9	**39**
10	**40**	36	27	35	36	35	38	9	**40**
11	**41**	38	28	37	36	36	39	9	**41**
12	**42**	40	29	38	36	37	40	9	**42**
13	**43**	41	30	40	36	38	41	9	**43**
14	**44**	43	31	41	36	39	42	9	**44**
15	**46**	44	32	43	36	40	43	9	45
16	**47**	45	33	44	36	41	44	27	46
17	**48**	47	34	46	36	42	45	27	47
18	**49**	48	35	47	36	43	46	27	48
19	**51**	49	36	48	36	44	47	27	49
20	**52**	51	37	50	36	45	48	27	50
21	**53**	52	38	51	36	46	49	27	51
22	**54**	53	39	52	36	47	50	27	52
23	**55**	54	40	53	36	48	51	27	53
24	**56**	**56**	41	55	36	49	52	27	54
25	**58**	57	42	56	36	50	53	27	55
26	**59**	58	43	57	36	51	54	27	56
27	**60**	**60**	44	59	36	52	55	27	57
28	**61**	**61**	45	60	36	53	56	27	58
29	**63**	62	46	61	36	54	57	27	59
30	**64**	63	47	62	36	55	58	27	60

Table A.8: Bounds for codes with $q = 2$, $d = 18$, $3 \leq k \leq 30$.

$q = 2, \quad d = 20$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	31	27	22	25	**35**	31	32	15	-
4	36	29	23	27	**38**	32	34	15	-
5	38	30	24	29	**39**	33	35	15	-
6	39	32	25	31	**40**	34	36	15	**40**
7	40	34	26	33	40	35	37	15	**41**
8	41	36	27	35	40	36	38	15	**42**
9	42	38	28	37	40	37	39	15	**43**
10	43	40	29	38	40	38	40	15	**44**
11	**45**	41	30	40	40	39	41	15	**45**
12	**46**	42	31	41	40	40	42	15	**46**
13	**47**	44	32	43	40	41	43	15	**47**
14	**48**	45	33	44	40	42	44	15	**48**
15	**50**	47	34	46	40	43	45	15	49
16	**51**	48	35	47	40	44	46	35	50
17	**52**	50	36	48	40	45	47	35	51
18	**53**	51	37	50	40	46	48	35	52
19	**54**	52	38	51	40	47	49	35	53
20	**56**	54	39	53	40	48	50	35	54
21	**57**	55	40	54	40	49	51	35	55
22	**58**	56	41	55	40	50	52	35	56
23	**59**	58	42	57	40	51	53	35	57
24	**61**	59	43	58	40	52	54	35	58
25	**62**	60	44	59	40	53	55	35	59
26	**63**	62	45	61	40	54	56	35	60
27	**64**	63	46	62	40	55	57	35	61
28	**65**	64	47	63	40	56	58	35	62
29	**66**	**66**	48	64	40	57	59	35	63
30	**67**	**67**	49	66	40	58	60	35	64

Table A.9: Bounds for codes with $q = 2$, $d = 20$, $3 \leq k \leq 30$.

$q = 2, \quad d = 22$									
k	**EB**	**JB**	**SB**	**HB**	**PB**	**bA**	**bB**	**bC**	**bD**
3	35	29	24	27	**39**	34	34	11	-
4	39	31	25	30	**42**	35	36	11	-
5	42	33	26	32	**43**	36	37	11	-
6	43	35	27	34	**44**	37	38	11	**44**
7	44	37	28	36	44	38	39	11	**45**
8	45	39	29	37	44	39	40	11	**46**
9	46	40	30	39	44	40	41	11	**47**
10	47	42	31	41	44	41	42	11	**48**
11	48	44	32	42	44	42	43	11	**49**
12	**50**	45	33	44	44	43	44	11	**50**
13	**51**	47	34	46	44	44	45	11	**51**
14	**52**	48	35	47	44	45	46	11	**52**
15	**53**	50	36	49	44	46	47	11	**53**
16	**55**	51	37	50	44	47	48	33	54
17	**56**	53	38	51	44	48	49	33	55
18	**57**	54	39	53	44	49	50	33	56
19	**58**	55	40	54	44	50	51	33	57
20	**60**	57	41	56	44	51	52	33	58
21	**61**	58	42	57	44	52	53	33	59
22	**62**	60	43	58	44	53	54	33	60
23	**63**	61	44	60	44	54	55	33	61
24	**64**	62	45	61	44	55	56	33	62
25	**66**	64	46	62	44	56	57	33	63
26	**67**	65	47	64	44	57	58	33	64
27	**68**	66	48	65	44	58	59	33	65
28	**69**	67	49	66	44	59	60	33	66
29	**71**	69	50	68	44	60	61	33	67
30	**72**	70	51	69	44	61	62	33	68

Table A.10: Bounds for codes with $q = 2$, $d = 22$, $3 \leq k \leq 30$.

A.2 Bounds for codes in $\mathbb{F}_3$

	$q = 3, \quad d = 6$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**9**	8	8	**9**	**9**	8	**9**
4	**10**	9	9	9	**10**	8	**10**
5	**11**	10	10	9	**11**	8	**11**
6	**12**	11	11	9	**12**	8	**12**
7	**13**	12	**13**	9	**13**	8	**13**
8	**14**	13	**14**	9	**14**	8	**14**
9	**15**	14	**15**	9	**15**	8	**15**
10	**16**	15	**16**	9	**16**	8	**16**
11	**18**	16	17	9	17	8	17

Table A.11: Bounds for codes with $q = 3$, $d = 6$, $3 \leq k \leq 11$.

	$q = 3, \quad d = 7$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	10	9	10	**11**	10	7	**11**
4	11	10	11	11	11	7	**12**
5	12	11	12	11	12	7	**13**
6	13	12	**14**	11	13	7	**14**
7	14	13	**15**	11	14	7	**15**
8	**16**	14	**16**	11	15	7	**16**
9	**17**	15	**17**	11	16	7	**17**
10	18	16	**19**	11	17	7	18
11	19	17	**20**	11	18	7	19

Table A.12: Bounds for codes with $q = 3$, $d = 7$, $3 \leq k \leq 11$.

$q = 3, \quad d = 8$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**12**	10	10	**12**	11	8	**12**
4	12	11	11	12	12	8	**13**
5	13	12	12	12	13	8	**14**
6	**15**	13	14	12	14	8	**15**
7	**16**	14	15	12	15	8	**16**
8	**17**	15	16	12	16	8	**17**
9	**18**	16	17	12	17	8	**18**
10	**19**	17	**19**	12	18	8	**19**
11	**20**	18	**20**	12	19	8	**20**
12	**21**	19	**21**	12	20	8	**21**

Table A.13: Bounds for codes with $q = 3$, $d = 8$, $3 \leq k \leq 12$.

$q = 3, \quad d = 9$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**13**	11	12	**13**	**13**	**13**	-
4	**14**	12	13	**14**	**14**	13	**14**
5	**15**	13	14	14	**15**	13	**15**
6	**16**	14	**16**	14	**16**	13	**16**
7	**17**	15	**17**	14	**17**	13	**17**
8	**18**	16	**18**	14	**18**	13	**18**
9	19	17	**20**	14	19	13	19
10	20	18	**21**	14	20	13	20
11	**22**	19	**22**	14	21	13	21
12	**23**	20	**23**	14	22	13	22

Table A.14: Bounds for codes with $q = 3$, $d = 9$, $3 \leq k \leq 12$.

$q = 3, \quad d = 10$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	14	12	12	**15**	14	10	-
4	**15**	13	13	**15**	**15**	10	**15**
5	**16**	14	14	15	**16**	10	**16**
6	**17**	15	16	15	**17**	10	**17**
7	**18**	16	17	15	**18**	10	**18**
8	**20**	17	18	15	19	10	19
9	**21**	18	20	15	20	10	20
10	**22**	19	21	15	21	10	21
11	**23**	20	22	15	22	10	22
12	**24**	21	23	15	23	10	23

Table A.15: Bounds for codes with $q = 3$, $d = 10$, $3 \leq k \leq 12$.

$q = 3, \quad d = 11$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**16**	13	13	**16**	15	11	-
4	**17**	14	15	**17**	16	11	**17**
5	**18**	15	16	17	17	11	**18**
6	**19**	16	18	17	18	11	**19**
7	**20**	17	19	17	19	11	**20**
8	**21**	18	**21**	17	20	11	**21**
9	**22**	19	**22**	17	21	11	**22**
10	**23**	20	**23**	17	22	11	**23**
11	**24**	21	**24**	17	23	11	**24**
12	25	22	**26**	17	24	11	25

Table A.16: Bounds for codes with $q = 3$, $d = 11$, $3 \leq k \leq 12$.

$q = 3, \quad d = 12$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	17	14	13	**18**	16	16	-
4	**18**	15	15	**18**	17	16	**18**
5	**19**	16	16	18	18	16	**19**
6	**20**	17	18	18	19	16	**20**
7	**21**	18	19	18	20	16	**21**
8	**22**	19	21	18	21	16	**22**
9	**23**	20	22	18	22	16	**23**
10	**25**	21	23	18	23	16	24
11	**26**	22	24	18	24	16	25
12	**27**	23	26	18	25	16	26

Table A.17: Bounds for codes with $q = 3$, $d = 12$, $3 \leq k \leq 12$.

$q = 3, \quad d = 13$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**19**	15	15	**19**	17	13	-
4	**20**	16	17	**20**	18	13	**20**
5	20	17	18	20	19	13	**21**
6	21	18	20	20	20	13	**22**
7	**23**	19	21	20	21	13	**23**
8	**24**	20	23	20	22	13	**24**
9	**25**	21	24	20	23	13	**25**
10	**26**	22	25	20	24	13	**26**
11	**27**	23	**27**	20	25	13	**27**
12	**28**	24	**28**	20	26	13	**28**

Table A.18: Bounds for codes with $q = 3$, $d = 13$, $3 \leq k \leq 12$.

A.3 Bounds for codes in $\mathbb{F}_4$

	$q = 4, \quad d = 9$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**12**	11	11	**12**	**12**	9	**12**
4	**13**	12	12	12	**13**	9	**13**
5	**14**	13	**14**	12	**14**	9	**14**
6	**15**	14	**15**	12	**15**	9	**15**
7	**16**	15	**16**	12	**16**	9	**16**
8	**17**	16	**17**	12	**17**	9	**17**
9	18	17	**19**	12	18	9	18
10	**20**	18	**20**	12	19	9	19
11	**21**	19	**21**	12	20	9	20

Table A.19: Bounds for codes with $q = 4$, $d = 9$, $3 \leq k \leq 11$.

	$q = 4, \quad d = 10$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	13	12	11	**14**	13	10	**14**
4	14	13	12	14	14	10	**15**
5	15	14	14	14	15	10	**16**
6	16	15	15	14	16	10	**17**
7	17	16	16	14	17	10	**18**
8	**19**	17	17	14	18	10	**19**
9	**20**	18	19	14	19	10	**20**
10	**21**	19	20	14	20	10	**21**
11	**22**	20	21	14	21	10	**22**

Table A.20: Bounds for codes with $q = 4$, $d = 10$, $3 \leq k \leq 11$.

$q = 4, \quad d = 11$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**15**	13	12	**15**	14	11	**15**
4	15	14	14	15	15	11	**16**
5	16	15	15	15	16	11	**17**
6	17	16	17	15	17	11	**18**
7	**19**	17	18	15	18	11	**19**
8	**20**	18	19	15	19	11	**20**
9	**21**	19	20	15	20	11	**21**
10	**22**	20	**22**	15	21	11	**22**
11	**23**	21	**23**	15	22	11	**23**
12	**24**	22	**24**	15	23	11	**24**

Table A.21: Bounds for codes with $q = 4$, $d = 11$, $3 \leq k \leq 12$.

$q = 4, \quad d = 12$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**16**	14	12	**16**	**16**	15	**16**
4	**17**	15	14	16	**17**	15	**17**
5	**18**	16	15	16	**18**	15	**18**
6	**19**	17	17	16	**19**	15	**19**
7	**20**	18	18	16	**20**	15	**20**
8	**21**	19	19	16	**21**	15	**21**
9	**22**	20	20	16	**22**	15	**22**
10	**23**	21	22	16	**23**	15	**23**
11	**24**	22	23	16	**24**	15	**24**
12	**25**	23	24	16	**25**	15	**25**

Table A.22: Bounds for codes with $q = 4$, $d = 12$, $3 \leq k \leq 12$.

$q = 4, \quad d = 13$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	17	15	14	**18**	17	13	**18**
4	18	16	16	18	18	13	**19**
5	19	17	17	18	19	13	**20**
6	20	18	18	18	20	13	**21**
7	21	19	20	18	21	13	**22**
8	22	20	21	18	22	13	**23**
9	23	21	22	18	23	13	**24**
10	24	22	24	18	24	13	**25**
11	25	23	25	18	25	13	**26**
12	26	24	26	18	26	13	**27**

Table A.23: Bounds for codes with $q = 4$, $d = 13$, $3 \leq k \leq 12$.

$q = 4, \quad d = 14$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**19**	16	14	**19**	18	14	**19**
4	19	17	16	19	19	14	**20**
5	20	18	17	19	20	14	**21**
6	21	19	18	19	21	14	**22**
7	22	20	20	19	22	14	**23**
8	23	21	21	19	23	14	**24**
9	**25**	22	22	19	24	14	**25**
10	**26**	23	24	19	25	14	**26**
11	**27**	24	25	19	26	14	**27**
12	**28**	25	26	19	27	14	**28**

Table A.24: Bounds for codes with $q = 4$, $d = 14$, $3 \leq k \leq 12$.

$q = 4, \quad d = 15$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**20**	17	16	**20**	19	15	**20**
4	**21**	18	17	20	20	15	**21**
5	21	19	19	20	21	15	**22**
6	22	20	20	20	22	15	**23**
7	**24**	21	22	20	23	15	**24**
8	**25**	22	23	20	24	15	**25**
9	**26**	23	24	20	25	15	**26**
10	**27**	24	26	20	26	15	**27**
11	**28**	25	27	20	27	15	**28**
12	**29**	26	28	20	28	15	**29**

Table A.25: Bounds for codes with $q = 4$, $d = 15$, $3 \leq k \leq 12$.

$q = 4, \quad d = 16$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**21**	18	16	**21**	**21**	**21**	-
4	**22**	19	17	**22**	**22**	21	**22**
5	**23**	20	19	22	**23**	21	**23**
6	**24**	21	20	22	**24**	21	**24**
7	**25**	22	22	22	**25**	21	**25**
8	**26**	23	23	22	**26**	21	**26**
9	**27**	24	24	22	**27**	21	**27**
10	**28**	25	26	22	**28**	21	**28**
11	**29**	26	27	22	**29**	21	**29**
12	**30**	27	28	22	**30**	21	**30**

Table A.26: Bounds for codes with $q = 4$, $d = 16$, $3 \leq k \leq 12$.

A.4 Bounds for codes in $\mathbb{F}_5$

	$q = 5, \quad d = 11$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**14**	13	12	**14**	**14**	11	**14**
4	**15**	14	13	14	**15**	11	**15**
5	**16**	15	15	14	**16**	11	**16**
6	**17**	16	16	14	**17**	11	**17**
7	**18**	17	17	14	**18**	11	**18**
8	**19**	18	18	14	**19**	11	**19**
9	**20**	19	**20**	14	**20**	11	**20**
10	**21**	20	**21**	14	**21**	11	**21**
11	**22**	21	**22**	14	**22**	11	**22**

Table A.27: Bounds for codes with $q = 5$, $d = 11$, $3 \leq k \leq 11$.

	$q = 5, \quad d = 12$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**15**	14	12	**15**	**15**	12	**15**
4	**16**	15	13	15	**16**	12	**16**
5	**17**	16	15	15	**17**	12	**17**
6	**18**	17	16	15	**18**	12	**18**
7	**19**	18	17	15	**19**	12	**19**
8	**20**	19	18	15	**20**	12	**20**
9	**21**	20	20	15	**21**	12	**21**
10	**22**	21	21	15	**22**	12	**22**
11	**24**	22	22	15	23	12	23

Table A.28: Bounds for codes with $q = 5$, $d = 12$, $3 \leq k \leq 11$.

$q = 5, \quad d = 13$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	16	15	13	**17**	16	13	**17**
4	17	16	15	17	17	13	**18**
5	18	17	16	17	18	13	**19**
6	19	18	18	17	19	13	**20**
7	20	19	19	17	20	13	**21**
8	21	20	20	17	21	13	**22**
9	22	21	22	17	22	13	**23**
10	**24**	22	23	17	23	13	**24**
11	**25**	23	24	17	24	13	**25**
12	**26**	24	25	17	25	13	**26**

Table A.29: Bounds for codes with $q = 5$, $d = 13$, $3 \leq k \leq 12$.

$q = 5, \quad d = 14$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**18**	16	14	**18**	17	14	**18**
4	18	17	15	18	18	14	**19**
5	19	18	16	18	19	14	**20**
6	20	19	18	18	20	14	**21**
7	21	20	19	18	21	14	**22**
8	**23**	21	20	18	22	14	**23**
9	**24**	22	22	18	23	14	**24**
10	**25**	23	23	18	24	14	**25**
11	**26**	24	24	18	25	14	**26**
12	**27**	25	25	18	26	14	**27**

Table A.30: Bounds for codes with $q = 5$, $d = 14$, $3 \leq k \leq 12$.

$q = 5, \quad d = 15$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**19**	17	15	**19**	**19**	18	**19**
4	**20**	18	16	19	**20**	18	**20**
5	**21**	19	18	19	**21**	18	**21**
6	21	20	19	19	**22**	18	**22**
7	**23**	21	21	19	**23**	18	**23**
8	**24**	22	22	19	**24**	18	**24**
9	**25**	23	23	19	**25**	18	**25**
10	**26**	24	25	19	**26**	18	**26**
11	**27**	25	26	19	**27**	18	**27**
12	**28**	26	27	19	**28**	18	**28**

Table A.31: Bounds for codes with $q = 5$, $d = 15$, $3 \leq k \leq 12$.

$q = 5, \quad d = 16$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**20**	18	16	**20**	**20**	16	**20**
4	**21**	19	16	20	**21**	16	**21**
5	**22**	20	18	20	**22**	16	**22**
6	**23**	21	19	20	**23**	16	**23**
7	**24**	22	21	20	**24**	16	**24**
8	**25**	23	22	20	**25**	16	**25**
9	**26**	24	23	20	**26**	16	**26**
10	**27**	25	25	20	**27**	16	**27**
11	**28**	26	26	20	**28**	16	**28**
12	**29**	27	27	20	**29**	16	**29**

Table A.32: Bounds for codes with $q = 5$, $d = 16$, $3 \leq k \leq 12$.

$q = 5, \quad d = 17$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	21	19	17	**22**	21	17	**22**
4	22	20	18	22	22	17	**23**
5	23	21	19	22	23	17	**24**
6	24	22	21	22	24	17	**25**
7	25	23	22	22	25	17	**26**
8	26	24	24	22	26	17	**27**
9	27	25	25	22	27	17	**28**
10	28	26	26	22	28	17	**29**
11	29	27	28	22	29	17	**30**
12	30	28	29	22	30	17	**31**

Table A.33: Bounds for codes with $q = 5$, $d = 17$, $3 \leq k \leq 12$.

$q = 5, \quad d = 18$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**23**	20	18	**23**	22	18	**23**
4	23	21	18	23	23	18	**24**
5	24	22	19	23	24	18	**25**
6	25	23	21	23	25	18	**26**
7	26	24	22	23	26	18	**27**
8	27	25	24	23	27	18	**28**
9	28	26	25	23	28	18	**29**
10	29	27	26	23	29	18	**30**
11	30	28	28	23	30	18	**31**
12	32	29	29	23	31	18	**32**

Table A.34: Bounds for codes with $q = 5$, $d = 18$, $3 \leq k \leq 12$.

A.5 Bounds for codes in $\mathbb{F}_7$

	$q = 7, \quad d = 15$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**18**	17	15	**18**	**18**	15	**18**
4	**19**	18	16	18	**19**	15	**19**
5	**20**	19	17	18	**20**	15	**20**
6	**21**	20	18	18	**21**	15	**21**
7	**22**	21	20	18	**22**	15	**22**
8	**23**	22	21	18	**23**	15	**23**
9	23	23	22	18	**24**	15	**24**
10	**25**	24	23	18	**25**	15	**25**
11	**26**	25	25	18	**26**	15	**26**

Table A.35: Bounds for codes with $q = 7$, $d = 15$, $3 \leq k \leq 11$.

	$q = 7, \quad d = 16$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**19**	18	16	**19**	**19**	16	**19**
4	**20**	19	16	19	**20**	16	**20**
5	**21**	20	17	19	**21**	16	**21**
6	**22**	21	18	19	**22**	16	**22**
7	**23**	22	20	19	**23**	16	**23**
8	**24**	23	21	19	**24**	16	**24**
9	**25**	24	22	19	**25**	16	**25**
10	**26**	25	23	19	**26**	16	**26**
11	**27**	26	25	19	**27**	16	**27**

Table A.36: Bounds for codes with $q = 7$, $d = 16$, $3 \leq k \leq 11$.

$q = 7, \quad d = 17$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**20**	19	17	**20**	**20**	17	**20**
4	**21**	20	17	20	**21**	17	**21**
5	**22**	21	18	20	**22**	17	**22**
6	**23**	22	20	20	**23**	17	**23**
7	**24**	23	21	20	**24**	17	**24**
8	**25**	24	22	20	**25**	17	**25**
9	**26**	25	24	20	**26**	17	**26**
10	**27**	26	25	20	**27**	17	**27**
11	**28**	27	26	20	**28**	17	**28**
12	**29**	28	27	20	**29**	17	**29**

Table A.37: Bounds for codes with $q = 7$, $d = 17$, $3 \leq k \leq 12$.

$q = 7, \quad d = 18$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**21**	20	18	**21**	**21**	18	**21**
4	**22**	21	18	21	**22**	18	**22**
5	**23**	22	18	21	**23**	18	**23**
6	**24**	23	20	21	**24**	18	**24**
7	**25**	24	21	21	**25**	18	**25**
8	**26**	25	22	21	**26**	18	**26**
9	**27**	26	24	21	**27**	18	**27**
10	**28**	27	25	21	**28**	18	**28**
11	**29**	28	26	21	**29**	18	**29**
12	**30**	29	27	21	**30**	18	**30**

Table A.38: Bounds for codes with $q = 7$, $d = 18$, $3 \leq k \leq 12$.

$q = 7, \quad d = 19$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**23**	21	19	**23**	22	19	**23**
4	23	22	19	23	23	19	**24**
5	24	23	20	23	24	19	**25**
6	25	24	21	23	25	19	**26**
7	26	25	23	23	26	19	**27**
8	27	26	24	23	27	19	**28**
9	28	27	25	23	28	19	**29**
10	29	28	27	23	29	19	**30**
11	30	29	28	23	30	19	**31**
12	31	30	29	23	31	19	**32**

Table A.39: Bounds for codes with $q = 7$, $d = 19$, $3 \leq k \leq 12$.

$q = 7, \quad d = 20$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**24**	22	20	**24**	23	20	**24**
4	24	23	20	24	24	20	**25**
5	25	24	20	24	25	20	**26**
6	26	25	21	24	26	20	**27**
7	27	26	23	24	27	20	**28**
8	28	27	24	24	28	20	**29**
9	29	28	25	24	29	20	**30**
10	30	29	27	24	30	20	**31**
11	31	30	28	24	31	20	**32**
12	32	31	29	24	32	20	**33**

Table A.40: Bounds for codes with $q = 7$, $d = 20$, $3 \leq k \leq 12$.

$q = 7, \quad d = 21$

k	EB	SB	HB	PB	bB	bC	bD
3	**25**	23	21	**25**	**25**	24	**25**
4	**26**	24	21	25	**26**	24	**26**
5	26	25	21	25	**27**	24	**27**
6	27	26	23	25	**28**	24	**28**
7	28	27	24	25	**29**	24	**29**
8	29	28	25	25	**30**	24	**30**
9	30	29	27	25	**31**	24	**31**
10	31	30	28	25	**32**	24	**32**
11	32	31	29	25	**33**	24	**33**
12	33	32	31	25	**34**	24	**34**

Table A.41: Bounds for codes with $q = 7$, $d = 21$, $3 \leq k \leq 12$.

$q = 7, \quad d = 22$

k	EB	SB	HB	PB	bB	bC	bD
3	**26**	24	22	**26**	**26**	22	**26**
4	**27**	25	22	26	**27**	22	**27**
5	**28**	26	22	26	**28**	22	**28**
6	28	27	23	26	**29**	22	**29**
7	29	28	24	26	**30**	22	**30**
8	30	29	25	26	**31**	22	**31**
9	31	30	27	26	**32**	22	**32**
10	32	31	28	26	**33**	22	**33**
11	33	32	29	26	**34**	22	**34**
12	**35**	33	31	26	**35**	22	**35**

Table A.42: Bounds for codes with $q = 7$, $d = 22$, $3 \leq k \leq 12$.

A.6 Bounds for codes in $\mathbb{F}_8$

	$q = 8, \quad d = 17$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**20**	19	17	**20**	**20**	17	**20**
4	**21**	20	17	20	**21**	17	**21**
5	**22**	21	18	20	**22**	17	**22**
6	**23**	22	19	20	**23**	17	**23**
7	23	23	21	20	**24**	17	**24**
8	24	24	22	20	**25**	17	**25**
9	**26**	25	23	20	**26**	17	**26**
10	**27**	26	25	20	**27**	17	**27**
11	27	27	26	20	**28**	17	**28**

Table A.43: Bounds for codes with $q = 8$, $d = 17$, $3 \leq k \leq 11$.

	$q = 8, \quad d = 18$						
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**21**	20	18	**21**	**21**	18	**21**
4	**22**	21	18	21	**22**	18	**22**
5	**23**	22	18	21	**23**	18	**23**
6	**24**	23	19	21	**24**	18	**24**
7	24	24	21	21	**25**	18	**25**
8	**26**	25	22	21	**26**	18	**26**
9	**27**	26	23	21	**27**	18	**27**
10	27	27	25	21	**28**	18	**28**
11	**29**	28	26	21	**29**	18	**29**

Table A.44: Bounds for codes with $q = 8$, $d = 18$, $3 \leq k \leq 11$.

$q = 8, \quad d = 19$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**22**	21	19	**22**	**22**	19	**22**
4	**23**	22	19	22	**23**	19	**23**
5	**24**	23	19	22	**24**	19	**24**
6	**25**	24	21	22	**25**	19	**25**
7	**26**	25	22	22	**26**	19	**26**
8	**27**	26	24	22	**27**	19	**27**
9	27	27	25	22	**28**	19	**28**
10	**29**	28	26	22	**29**	19	**29**
11	**30**	29	27	22	**30**	19	**30**
12	**31**	30	29	22	**31**	19	**31**

Table A.45: Bounds for codes with $q = 8$, $d = 19$, $3 \leq k \leq 12$.

$q = 8, \quad d = 20$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**23**	22	20	**23**	**23**	20	**23**
4	**24**	23	20	23	**24**	20	**24**
5	**25**	24	20	23	**25**	20	**25**
6	**26**	25	21	23	**26**	20	**26**
7	**27**	26	22	23	**27**	20	**27**
8	**28**	27	24	23	**28**	20	**28**
9	**29**	28	25	23	**29**	20	**29**
10	**30**	29	26	23	**30**	20	**30**
11	**31**	30	27	23	**31**	20	**31**
12	**32**	31	29	23	**32**	20	**32**

Table A.46: Bounds for codes with $q = 8$, $d = 20$, $3 \leq k \leq 12$.

$q = 8, \quad d = 21$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**24**	23	21	**24**	**24**	21	**24**
4	**25**	24	21	24	**25**	21	**25**
5	**26**	25	21	24	**26**	21	**26**
6	**27**	26	22	24	**27**	21	**27**
7	**28**	27	24	24	**28**	21	**28**
8	**29**	28	25	24	**29**	21	**29**
9	**30**	29	26	24	**30**	21	**30**
10	**31**	30	28	24	**31**	21	**31**
11	**32**	31	29	24	**32**	21	**32**
12	**33**	32	30	24	**33**	21	**33**

Table A.47: Bounds for codes with $q = 8$, $d = 21$, $3 \leq k \leq 12$.

$q = 8, \quad d = 22$							
k	**EB**	**SB**	**HB**	**PB**	**bB**	**bC**	**bD**
3	**26**	24	22	**26**	25	22	**26**
4	26	25	22	26	26	22	**27**
5	27	26	22	26	27	22	**28**
6	28	27	22	26	28	22	**29**
7	29	28	24	26	29	22	**30**
8	30	29	25	26	30	22	**31**
9	31	30	26	26	31	22	**32**
10	32	31	28	26	32	22	**33**
11	33	32	29	26	33	22	**34**
12	34	33	30	26	34	22	**35**

Table A.48: Bounds for codes with $q = 8$, $d = 22$, $3 \leq k \leq 12$.

$q = 8, \quad d = 23$

k	EB	SB	HB	PB	bB	bC	bD
3	**27**	25	23	**27**	26	23	**27**
4	27	26	23	27	27	23	**28**
5	28	27	23	27	28	23	**29**
6	29	28	24	27	29	23	**30**
7	30	29	25	27	30	23	**31**
8	31	30	26	27	31	23	**32**
9	32	31	28	27	32	23	**33**
10	33	32	29	27	33	23	**34**
11	34	33	30	27	34	23	**35**
12	35	34	32	27	35	23	**36**

Table A.49: Bounds for codes with $q = 8$, $d = 23$, $3 \leq k \leq 12$.

$q = 8, \quad d = 24$

k	EB	SB	HB	PB	bB	bC	bD
3	**28**	26	24	**28**	**28**	27	**28**
4	**29**	27	24	28	**29**	27	**29**
5	29	28	24	28	**30**	27	**30**
6	30	29	24	28	**31**	27	**31**
7	31	30	25	28	**32**	27	**32**
8	32	31	26	28	**33**	27	**33**
9	33	32	28	28	**34**	27	**34**
10	34	33	29	28	**35**	27	**35**
11	35	34	30	28	**36**	27	**36**
12	36	35	32	28	**37**	27	**37**

Table A.50: Bounds for codes with $q = 8$, $d = 24$, $3 \leq k \leq 12$.

Bibliography

[1] R. D. Baker, J. H. Van Lint, and R. M. Wilson. On the Preparata and Goethals codes. *IEEE Transactions on Information Theory*, 29(3):342–345, 1983.

[2] E. B. Barker and J. M. Kelsey. SP 800-90b. Recommendation for the entropy sources used for random bit generation - DRAFT. *US Department of Commerce, National Institute of Standards and Technology*, 2012.

[3] E. B. Barker and J. M. Kelsey. SP 800-90c. recommendation for random bit generator (rbg) constructions - DRAFT. *US Department of Commerce, National Institute of Standards and Technology*, 2012.

[4] E. B. Barker and J. M. Kelsey. SP 800-90a. Recommendation for Random Number Generation using deterministic Random Bit Generators - Revision 1. *US Department of Commerce, National Institute of Standards and Technology*, 2015.

[5] L. A. Bassalygo. New upper bounds for error correcting codes. *Problemy Peredachi Informatsii*, 1(4):41–44, 1965.

[6] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, et al. SP 800-22 rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic

Applications. *US Department of Commerce, National Institute of Standards and Technology*, 2000.

[7] E. Bellini, E. Guerrini, and M. Sala. Some bounds on the size of codes. *IEEE Transactions on Information Theory*, 60(3):1475–1480, 2014.

[8] A. Bonisoli. Every equidistant linear code is a sequence of dual hamming codes. *Ars Combinatoria*, 18(2):181–186, 1984.

[9] C. Carlet. Boolean functions for cryptography and error correcting codes. *Boolean models and methods in mathematics, computer science, and engineering*, 2:257, 2010.

[10] P. Davis. Circulant matrices. *AMS Chelsea Publishing Series*, 1994.

[11] P. Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Research Reports Supplements*, (10): vi+97, 1973.

[12] E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31(3):504–522, 1952.

[13] M. Golay. Notes on Digital Coding. *Proceedings IRE*, 37:657, 1949.

[14] J. H. Griesmer. A bound for error-correcting codes. *IBM Journal of Research and Development*, 4(5):532–542, 1960.

[15] E. Guerrini. *Systematic codes and polynomial ideals*. PhD thesis, University of Trento, 2009.

[16] E. Guerrini, A. Meneghetti, and M. Sala. On optimal nonlinear systematic codes. *IEEE Transactions on Information Theory*, 62 (6):3103–3112, 2016.

[17] N. Hamada. A characterization of some [n, k, d; q]-codes meeting the Griesmer bound using a minihyper in a finite projective geometry. *Discrete Mathematics*, 116(1):229–268, 1993.

[18] N. Hamada and T. Helleseth. A characterization of some ternary codes meeting the Griesmer bound. *Finite Fields: Theory, Applications, and Algorithms*, 168:139–150, 1993.

[19] R. W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

[20] T. Helleseth. A characterization of codes meeting the Griesmer bound. *Information and Control*, 50(2):128–159, 1981.

[21] T. Helleseth. Projective codes meeting the Griesmer bound. *Discrete mathematics*, 106:265–271, 1992.

[22] R. Hill. A first course in coding theory. *Clarendon Press Oxford*, 1986.

[23] W. C. Huffman and V. Pless. Fundamentals of error-correcting codes. *Cambridge University Press*, 2003.

[24] S. Johnson. A new upper bound for error-correcting codes. *IRE Transactions on Information Theory*, 8(3):203–207, 1962.

[25] S. Johnson. On upper bounds for unrestricted binary-error-correcting codes. *IEEE Transactions on Information Theory*, 17 (4):466–478, 1971.

[26] A. M. Kerdock. A class of low-rate nonlinear binary codes. *Information and Control*, 20:182–187; ibid. **21** (1972), 395, 1972.

[27] A. Klein. On codes meeting the Griesmer bound. *Discrete Mathematics*, 274(1–3):289–297, 2004.

[28] P. Lacharme. Post-processing functions for a biased physical random number generator. In *Proceedings of FSE 2008*, volume 4593 of *LNCS*, pages 334–342, 2008.

[29] P. Lacharme. Analysis and construction of correctors. *IEEE Transactions on Information Theory*, 55(10):4742–4748, 2009.

[30] T. Laihonen and S. Litsyn. On upper bounds for minimum distance and covering radius of non-binary codes. *Designs, Codes and Cryptography*, 14(1):71–80, 1998.

[31] V. I. Levenshtein. The application of Hadamard matrices to a problem in coding. *Problems of Cybernetics*, (5):166–184, 1964.

[32] R. Lidl and H. Niederreiter. Finite Fields. *Encyclopedia of Mathematics and its Applications. Cambridge University Press*, 1997.

[33] F. J. MacWilliams and N. J. A. Sloane. The theory of error-correcting codes. I and II. *North-Holland Publishing Co.*, 1977.

[34] T. Maruta. On the non-existence of linear codes attaining the Griesmer bound. *Geometriae Dedicata*, 60(1):1–7, 1996.

[35] T. Maruta. On the Achievement of the Griesmer Bound. *Designs, Codes and Cryptography*, 12(1):83–87, 1997.

[36] R. E. Paley. On orthogonal matrices. *Journal of Mathematics and Physics*, 12(1):311–320, 1933.

[37] J. Pearl. Application of walsh transform to statistical analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):111–119, 1971.

[38] V. Pless, R. A. Brualdi, and W. C. Huffman. Handbook of coding theory. *Elsevier Science Inc.*, 1998.

[39] M. Plotkin. Binary Codes with Specified Minimum Distance. *IEEE Transactions on Information Theory*, 6(4):445–450, 1960.

[40] F. P. Preparata. A class of optimum nonlinear double-error correcting codes. *Information and Control*, 13(13):378–400, 1968.

[41] J. S. Rosenthal. Convergence rates for markov chains. *Siam Review*, 37(3):387–405, 1995.

[42] R. Roth. Introduction to coding theory. *Cambridge University Press*, 2006.

[43] R. Shaltiel. An introduction to randomness extractors. In *International Colloquium on Automata, Languages, and Programming*, pages 21–41. Springer, 2011.

[44] R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.

[45] G. Solomon and J. J. Stiffler. Algebraically punctured cyclic codes. *Information and Control*, 8(2):170–179, 1965.

[46] F. Tamari. On linear codes which attain the Solomon-Stiffler bound. *Discrete Mathematics*, 49(2):179–191, 1984.

[47] F. Tamari. A construction of some [n, k, d; q]-codes meeting the Griesmer bound. *Discrete Mathematics*, 116(1–3):269–287, 1993.

[48] A. Tomasi, A. Meneghetti, and M. Sala. Code generator matrices as entropy extractors. *arXiv preprint arXiv:1502.01494*, 2015.

[49] H. Van Tilborg. On the uniqueness resp. nonexistence of certain codes meeting the Griesmer bound. *Information and control*, 44 (1):16–35, 1980.

[50] R. Varshamov. Estimate of the number of signals in error correcting codes. In *Dokl. Akad. Nauk SSSR*, volume 117, pages 739–741, 1957.

[51] H. N. Ward. Divisibility of codes meeting the Griesmer bound. *Journal of Combinatorial Theory, Series A*, 83(1):79–93, 1998.

[52] H. Zhou and J. Bruck. Linear extractors for extracting randomness from noisy sources. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 1738–1742. IEEE, 2011.

[53] V. A. Zinov'ev and S. N. Litsyn. On Shortening of Codes. *Problemy Peredachi Informatsii*, 20(1):3–11, 1984.