Error correcting codes

José Felipe Voloch

Chapter I

GENERALITIES

Denote by $\mathbf{F}_q$ the finite field with q elements. A linear code C over the alphabet $\mathbf{F}_q$ is a vector subspace of $\mathbf{F}_q^n$. If d is the dimension of C over $\mathbf{F}_q$ we say that C is a $[n, d]$-code.

There are various ways of describing a code , we can for example give a basis $v_1, \ldots, v_d$ of C. In this case , if $v_i = (v_{i1}, \ldots, v_{in})$ then the map $V : \mathbf{F}_q^d \to \mathbf{F}_q^n$ given by

$$V(a_1, \ldots, a_d) = \sum_{i=1}^{d} a_i v_i = (\sum_{i=1}^{d} a_i v_{i1}, \ldots, \sum_{i=1}^{d} a_i v_{in})$$

is an encoder, that is, regarding $\mathbf{F}_q^d$ as the set of words in a natural language , the function V tells how to encode the words.

The matrix $V = (v_{ij})$ that describes the linear map V is called a generating matrix of the code. We say that V is in standard form if $V = (I_d P)$ for a matrix P, that is $v_{ij} = \delta_{ij}$, for $i, j = 1, \ldots, d$ (where $\delta_{ij} = 0, i \neq j, \delta_{ii} = 1$). In this case we say that the first d symbols or coordinates of $c \in C$ are the information symbols and the remainder the check symbols.

Why check symbols? Given the vector $x = (x_1, \ldots, x_n) \in \mathbf{F}_q^n$ to check if $x \in C$ it suffices to check if $x = V(a)$ for some a and in this case (we are supposing C standard) $x_j = a_j, j \leq d$ and

$$x_j = \sum_{i=1}^{d} a_i v_{ij} = \sum_{i=1}^{d} x_i v_{ij}, j > d.$$

Hence, to check if $x \in C$ it is enough to verify whether $x_j = \sum_{i=1}^{d} x_i v_{ij}$, for $j = d+1, \ldots, n$.

We say that two codes C_1, C_2 are equivalent if we can obtain C_2 from C_1 by permuting coordinates. One can prove that every code is equivalent to a code that can be generated by a matrix in standard form (see ex. 9).

From what we saw above, the information contained in a word $c \in C$ depends on d of its coordinates and the rest is redundance that is used for control. Define then

the information rate of C, denoted by $i(C)$, as being $i(C) = d/n$. This measures the ratio between the number of coordinates carrying information and the total number of coordinates.

Another way of describing the code is to give a linear map $H : \mathbf{F}_q^n \to \mathbf{F}_q^{n-d}$ such that the kernel of H is C. In this case we have $(x_1, \ldots, x_n) \in C$ if and only if $H(x_1, \ldots, x_n) = 0$. If $H = (h_{ij})$ then the previous equation becomes

$$\sum_{j=1}^{n} h_{ij} x_j = 0, i = 1, \ldots, n-d.$$

The matrix H is then called the parity check matrix of of C.

If C is given by the generating matrix $(I_d P)$ as above, that is, C is standard, it is easy to calculate the check matrix of C. In fact we have $x \in C$ if and only if $x_j = \sum_{i=1}^{d} x_i v_{ij}, j > d$ as seen above, so the check matrix of C is $(-P^* I_{n-d})$ where P^* is the transpose of P.

An illustrative example, which originates the name parity check is the parity check code over $\mathbf{F}_2$ defined by the generating matrix $(I_n \mathbf{1})$ where $\mathbf{1} = (1, \ldots, 1)^*$ That is, $C \subset \mathbf{F}_q^{n+1}$ given by

$$\{(x_1, \ldots, x_n, x_1 + \cdots + x_n) \mid (x_1, \ldots, x_n \in \mathbf{F}_q^n\}.$$

The parity check matrix of C is $(1, \ldots., 1)$. It is easy to see then that $x \in C$ if and only if $\sum x_i = 0$. That is, x has an even number of non-zero coordinates, hence the name parity check.

Speaking of errors, we mentioned in the introduction that the codes would be chosen of such way that two words of the code were always very different, so that when we receive the message with possible errors we could decode it as the codeword most similar with receiveed message. To formalize the concepts of different and similar define the Hamming norm in $\mathbf{F}_q^n$ putting for $x \in \mathbf{F}_q^n$, $|x|$ = number of nonzero coordinates of x. Then the following properties hold:

1) $|x| = 0$ if and only if $x = 0$

2) $|\lambda x| = |x|$ if $\lambda \in \mathbf{F}_q, \lambda \neq 0$.

3) $|x + y| \leq |x| + |y|$.

We define the Hamming distance by $d(x,y) = |x-y|$, which satisfies

1') $d(x,y) = 0$ if and only if $x = y$

2') $d(x,y) = d(y,x)$

3') $d(x,z) \leq d(x.y) + d(y,z)$.

Note that $d(x,y)$ is the number of coordinates where x and y differ, hence $d(x,y)$ measures how different x and y are. The function d is a metric in $\mathbf{F}_q^n$.

Note that 1') follows from 1), 2') from 2) with $\lambda = -1$ and 3') from 3). Properties 1) and 2) are immediate. Let's prove now property 3).

Let $I = \{i \in \{1,\ldots,n\} \mid x_i = 0\}$ and $J = \{i \in \{1,\ldots,n\} \mid y_i = 0\}$ then, by definition, $|x| = n - \#I, |y| = n - \#J$. Hence

$$|x| + |y| = 2n - (\#I + \#J).$$

On the other hand, we have that $\#I + \#J = \#(I \cup J) + \#(I \cap J)$ and $\#(I \cup J) \leq n$, hence

$$|x| + |y| \geq n - \#(I \cap J).$$

However, if $i \in I \cap J, x_i = y_i = 0$, hence $x_i + y_i = 0$. Hence $x + y$ has i-th coordinate zero for all $i \in I \cap J$ consequently $|x+y| \leq n - \#(I \cap J)$. This concludes the proof.

This allows us to measure by how much words of the code differ among each other. We define the weight of the code C, denoted $w(c)$ putting $w(C) = \min\{d(x.y) \mid x, y \in C, x \neq y\}$. As C is a vector space, $x - y \in C$ whenever $x, y \in C$, thus $w(C) = \min\{|x| \mid x \in C, x \neq 0\}$.

[NEEDS IMPROVING]

We can now move on to correction and detection of errors . We say that the code C corrects e errors if for all $y \in \mathbf{F}_q^n$ there exists at most one $x \in C$ with $d(x,y) \leq e$. This means that when we receive the message y with at most e errors, that is, y differs from some element x of C in at most n coordinates, this element x being the message sent, then x is the unique element of C closest to y hence we can recover x from y. Later we will

discuss how to actually implement error correction. First we will discuss how many errors can a code correct.

Theorem 1. *Let C be a code of weight $w(C)$ then C corrects $[(w(C)-1)/2]$ errors.*

Proof: Let $e = [(w(C)-1)/2]$ and assume that C does not correct e errors and let $y \in \mathbf{F}_q^n$ such that there exists $x_1, x_2 \in C, x_1 \neq x_2$ with $d(x_i, y) \leq e, i = 1, 2$. From 3') we get $d(x_1, x_2) \leq d(x_1, y) + d(y, x_2) \leq 2e$. On the other hand, as $x_1 \neq x_2$, by definition of $w(C)$ we have $d(x_1, x_2) \geq 2e + 1$, contradiction.

A result that can be used to determine $w(C)$ is the following:

Proposition. *Let C be a code with check matrix H and weight $w(C)$. Then any $w(C)-1$ columns of H are linearly independent and there exists $w(C)$ linearly dependent columns of H.*

Proof: Let s be the integer such that any s columns of H are linearly independent and there exists $s+1$ linearly dependent columns of H.

Let $h_1, \ldots, h_n$ be the columns of H. If $h_{i_1}, \ldots, h_{i_{s+1}}$ are linearly dependent there exists $c_{i_1}, \ldots, c_{i_{s+1}} \in \mathbf{F}_q$ with $\sum c_{i_j} h_{i_j} = 0$. If $c = (c_1, \ldots, c_n)$ is defined by $c_i = c_{i_j}, i = i_j, c_i = 0$ otherwise, then $\sum c_i h_i = 0$ and $c \in C$. However c has at most $s+1$ nonzero coordinates, hence $w(C) \leq s+1$.

If $w(C) < s+1$, there exists $c \in C, c \neq 0$, with at most s coordinates not zero , say $c_i = 0$ if $i \neq i_1, \ldots, i_s$. As $c \in C$, we have $\sum c_i h_i = 0$, hence $\sum c_{i_j} h_{i_j} = 0$, hence $h_{i_1}, \ldots, h_{i_s}$ are linearly dependent, this contradicts the definition of s hence $w(C) \geq s+1$, as was to be proved.

Corollary (Singleton). *If C is a $[n, d]$ code then $w(C) \leq n - d + 1$.*

Proof: If H is the check matrix of C then the columns of H are in $\mathbf{F}_q^{n-d}$, hence any $n-d+1$ columns of H are linearly dependent. The result now follows from the proposition.

Codes with $w(C) = n - d + 1$ are called maximum distance separable codes (MDS). They have an interesting description as sets satisfying certain geometrical properties in projective spaces over finite fields that we will discuss in chapter IV.

We will now give a simple procedure for decoding.

If C is a $[n, d]$ code given as the kernel of $H : \mathbf{F}_q^n \to \mathbf{F}_q^{n-d}$, and $x \in \mathbf{F}_q^n$, we define the syndrome of x to be $H(x)$. For each $v \in \mathbf{F}_q^{n-d}$, choose $e_v \in \mathbf{F}_q^n$ such that $H(e_v) = v$ and such that $|e_v|$ is minimal on $H^{-1}(v)$. Such an e_v is called a coset leader of the coset $H^{-1}(v)$. There could be more than one suitable e_v but we fix a choice. If we receive the message y we compute $H(y) = v$ and take $c = y - e_v$ as the decoding of y.

Note that $H(c) = H(y) - H(e_v) = v - v = 0$, hence $c \in C$. Note also that $d(c, y) = |e_v|$. As e was chosen to minimize the Hamming norm on $H^{-1}(v)$, we have that c is the element of C closest to y. Consequently, if C corrects e errors, the decoding will give us the original message sent as long as the received message have syndrome v satisfying $|e_v| \leq e$. This process is called maximum likehood decoding.

In general this procedure is very costly because we need to calculate the e_v. Codes with special properties may have more efficient decoding algorithms. We will see some examples in later chapters.

Some examples:

1. Consider the [7, 4]-code over $\mathbf{F}_2$ with parity check matrix

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

This code is called the [7, 4]-Hamming code. It has weight 4 and a nice decoding algorithm.

[ADD STUFF HERE]

[DO HEXACODE]

Exercises:

1.??

2. Consider the [4, 2]-code over $\mathbf{F}_3$ with generator matrix

$$\begin{pmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 1 \end{pmatrix}$$

a) Compute the weight of C.

b) Compute a check matrix for C.

c) Decode $(0,1,1,1),(1,1,1,0)$ and $(0,0,2,2)$.

3. Let C, C' respectively be $[n,d]$ and $[n',d']$ codes. Consider the code

$$C \oplus C' = \{(c,c') \in \mathbf{F}_q^{n+n'} \mid c \in C, c' \in C'\}.$$

Show that $C \oplus C'$ is a $[n+n', d+d']$-code and that $w(C \oplus C') = w(C) + w(C')$. Find generating and check matrices for $C \oplus C'$ from given generating and check matrices for C and C'.

4. Let C be an $[n,d]$code. Let C' be the code consisting of $c \in C$ such that $c_{i_1} = \cdots = c_{i_l} = 0$. Consider, in the natural way, C as a code in $\mathbf{F}_q^{n-l}$. Show that $i_1, \ldots, i_l$ can be chosen such that $\dim C' = d - l$ and $w(C') = w(C)$.

5.Let C be an $[n,d]$code. If $i_1, \ldots, i_r \in \{1, \ldots, n\}$ consider, $A : \mathbf{F}_q^n \to \mathbf{F}_q^r, A(x_1, \ldots, x_n) =$ $(x_{i_1}, \ldots, x_{i_r})$. Let C' be the image of C by A. Prove that, if $r \leq n - d$, one can choose $i_1, \ldots, i_r$ such that C' is a $[r,d]$-code of weight $w(C') = w(C) - n + r$r.

6. Define for $x, y \in \mathbf{F}_q^n$ the inner product $(x,y) = \sum_{i=1}^n x_i y_i$. (Careful, this inner product Interno has nothing to do with the norm.) Give an example of $x \in \mathbf{F}_q^2$ with $x \neq 0$ and $(x,x) = 0$. If C is a code in $\mathbf{F}_q^n$ define the dual code as

$$C^\perp = \{y \in \mathbf{F}_q^n \mid (x,y) = 0, \forall x \in C\}.$$

What is the relation between the generating and check matrices of C and $C^\perp$? Give an example of a code C with $C = C^\perp$. Find $C^\perp$ for C as in exercise 2.

Note : The relation between the weights of C and $C^\perp$ in general is not simple. We have the following result due to MacWilliams (see eg [?]). Define, for a code C, the weight enumerator

$$P_C(t) = \sum_{i=0}^n \#\{c \in C \mid |c| = i\} t^i.$$

Then

$$P_{C^\perp}(t) = q^{-d}(1 + (q-1)t)^n P_C((1-t)/(1-(q-1)t).$$

7. Let C be a code with $w(C) = 2m$, we know that C corrects $m-1$ errors. Prove that C detects m errors, that is, if we receive a message, we can detect whether it has at most m errors and if it has at most m errors we can tell how many errors it has.

8. Magnetic tape for data storage in computers are usually recorded with nine heads, one on top of the other. That is, the data is stored in a matrix (b_{ij}), $i = 1, \ldots, 9$, $j = 1, \ldots, n, b_{ij} \in \mathbf{F}_2$, for some $n > 1$. For ease of machine reading it is required that there is an "on" bit in every column, This is guaranteed by requiring that $\sum_{i=1}^{9} b_{ij} = 1$ for all j (justify), Show also that this permits the correctioni of errors resulting of defects in one of the heads (a common occurrence). In order to correct other kinds of errors one can require also that $\sum_{j=1}^{n} b_{ij} = n (\bmod 2)$, for every i. To analize this code we may switch zeros and ones (ie set $c_{ij} = b_{ij} + 1$ for all i, j. The equations become then $\sum_{i=1}^{9} c_{ij} = \sum_{j=1}^{n} c_{ij} = 0$. Compute the dimension and weight of this code.

9. Prove that every code is equivalent to a code in standard form.

Chapter II

BOUNDS

In this this chapter we address the question of estimating a priori how good can we make a code. As this question is not fully solved we content ourselves in producing some bounds. We have already proved such a result, the Singleton bound saying that an $[n, d]$ code has weight at most $n - d + 1$. Our bounds will have similar form, giving restrictions on the weight of $[n, d]$ codes. We will also prove a result in another direction, namely the Varshamov-Gilbert bound which proves the existence of codes whose weight satisfies some inequality.

Let's introduce some notation.

If $r \leq n$ is an integer and $a \in \mathbf{F}_q^n$, we define the ball of center a and radius r as the set

$$B(a, r) = \{\chi \in \mathbf{F}_q^n \mid d(x, a) \leq r\}.$$

Let also $V_q(n, r) = \#B(a, r) = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i$.

Note that $\#B(a,r)$ does not depend on a since $x \mapsto x-a$ is a bijection between $B(a,r)$ and $B(0,r)$. The formula itself follows from the fact that to choose x of weight i, one needs to choose first which coordinates will nonzero, and there are $\binom{n}{i}$ ways of doing that, and second, which values those i coordinates will take, and there are $(q-1)^i$ ways of doing that.

If n, w are integers with $w \leq n$, define

$$A_q(n.w) = \max\{\dim C \mid C \subset \mathbf{F}_q^n \text{linearcode}, w(C) = w\}.$$

This is the largest possible dimension of a code of length n and weight w over $\mathbf{F}_q$.

Our first bound is the sphere-packing or Hamming bound.

Theorem (Hamming).

$$A_q(n,w) \leq \log_q(q^n/V_q(n,[(w-1)/2])).$$

Proof: Let C be an $[n,d]$code of weight w and put $e = [(w-1)/2]$. We saw in the last chapter that C corrects e errors, that is, the balls $B(c,e)$ are disjoint for $c \in C$. Hence,

$$\sum_{c \in C} \#B(c,e) \leq \#\mathbf{F}_q^n = q^n.$$

Therefore, $\#C.V_q(n,e) \leq q^n$ and, since $\#C = q^d$, the result follows.

As we saw in the proof the result follows by considering how many disjoint balls $B(c,e)$ we can pack in $\mathbf{F}_q^n$. This is related with the classical lattice sphere packing problem in Euclidean space. The interested reader may consult [21] and [??].

The technique of packing balls leads to the following result in the opposite direction.

Theorem (Varshamov-Gilbert). *If $w \leq n$ then*

$$A_q(n,w) \geq \log_q(q^n/V_q(n,w-1)).$$

Proof: The main idea is to keep packing balls as long as we have room. Let d be an integer smaller than $\log_q(q^n/V_q(n,w-1))$ and suppose that C is an $[n,d]$code of weight w.

We will show that we can find a code $C' \supset C$ of dimension $d+1$ and weight w. The result will then follow by starting with a one dimensional code spanned by an element of weight w and repeatedly increasing the code as long as $d < \log_q(q^n/V_q(n, w-1))$. We have that

$$\sum_{c \in C} \#B(c, w-1) = q^d V_q(n, w-1) < q^n = \#\mathbf{F}_q^n.$$

Thus, there exists $x \in \mathbf{F}_q^n, x \notin \cup_{c \in C} B(c, w-1)$. In particular, $x \notin C$ and $d(x, c) \geq w$ for all $c \in C$. Let C' be the code spanned by C and x, so that C' has dimension $d+1$.

To compute $w(C')$, let $c' \in C'$. Thus $c' = c + \lambda x, c \in C, \lambda \in \mathbf{F}_q$. If $\lambda = 0$ then $|c'| = |c| \geq w$, by hypothesis. If $\lambda \neq 0$ then

$$|c'| = |\lambda^{-1} c'| = |x + \lambda^{-1} c| = d(x, -c/\lambda) \geq w,$$

as follows from the construction of x. Thus $w(C') \geq w$ but as $C' \supset C$, we must have $w(C') = w$ and this completes the proof.

Although the theorem gives a construction of good codes in principle, the method is impractical. Later will discuss some constructions that produce good codes.

Returning to the problem of upper bounds for $A_q(n, w)$, we have

Theorem (Plotkin). *Set $\theta = 1 - 1/q$. If $w > n\theta$, then*

$$A_q(n, w) \leq \log_q(w/(w - n\theta)).$$

Proof: Let C be an $[n, d]$code of weight $w > n\theta$. Let S_i be the set $\{c = (c_1, \ldots, c_n) \in C \mid c_i \neq 0\}$. If $c \in C$, then c belongs to $|c|$ of the S_i's, thus $\sum_{c \in C} |c| = \sum_{i=1}^n \#S_i$.

On the other hand $C \setminus S_i$ is the linear subspace of C given by $c_i = 0$ which is either of codimension one in C or the whole of C. Therefore $\#S_i = 0$ or $\#S_i = q^d - q^{d-1} = \theta q^d$. Hence $\sum_{c \in C} |c| \leq n\theta q^d$.

We will now estimate $\sum_{c \in C} |c|$ from below. Note that, if $c \in C, c \neq 0$ we have $|c| \geq w$, so $\sum_{c \in C} |c| \geq w(q^d - 1)$. Putting the two estimates together gives $q^d \leq w/(w - n\theta)$ which proves the theorem.

We would like to compare those bounds and better understand them. For this purpose is covenient to look at them asymptotically. So we ask the question: what is the information rate for a code given its error-correcting rate, as the length gets larger? The answer is the number $\alpha_q(\delta) = \limsup A_q(n, [n\delta])$, where $\delta \in [0, 1]$ is the given error-correcting rate.

Manin has shown [13] that α_q is a continuous and decreasing function on $[0, 1]$. We will not prove it here.

Singleton's bound yields $\alpha_q(\delta) \leq 1 - \delta$ and Plotkin's bound gives $\alpha_q(\delta) \leq 0, \delta \geq \theta$, so $\alpha_q(\delta) = 0, \delta \geq \theta$.

Proposition. $\alpha_q(\delta) \leq 1 - \delta/\theta, 0 \leq \delta \leq \theta$.

Proof: Let C be an $[n, d]$code of weight $w \leq n\theta$. Let $m = [(w-1)/\theta]$ and consider the subcode C' of C given by $C' = \{c = (c_1, \ldots, c_n) \in C \mid c_i = 0, i \leq n - m\}$. We can project C' to $\mathbf{F}_q^m$ by considering the last m coordinates. We thus obtain a code of length m and dimension $k \geq d + m - n$. Moreover, since C' is a subcode of C, we have $w(C') \geq w$ and the same holds for the projected code. As $m\theta \leq w - 1 < w$ we can apply Plotkin's bound to the projected code and get

$$d + m - n \leq k \leq log_q(w/(w - m\theta)$$

.

Suppose now n is going to infinity and $w/n \to \delta, d/n \to \alpha$. Then $m \to \delta/\theta$, so the left hand side of the last inequality divided by n converges to $\alpha + \delta/\theta - 1$, whereas the right hand side of the last inequality divided by n converges to zero, yielding the proposition.

Hamming's bound gives $\alpha_q(\delta) \leq 1 - H_q(\delta/2), \delta \leq \theta$, where $H_q(x) = x \log \theta - x \log_q x - (1 - x) \log_q(1 - x), 0 < \delta \leq \theta$, and $H_q(0) = 0$. This follows from the following fact, which is a consequece of Stirling's formula,

$$\lim_{n \to \infty} \frac{\log_q V_q(n, [\lambda n])}{n} = H_q(\lambda).$$

Singleton's bound is worse than the other two bounds mentioned above. Hamming's bound beats proposition ? only in some subinterval $[0, \delta_0]$ of $[0, \delta]$. The best known bound

due to Elias, is the following,

$$\alpha_q(\delta) \leq 1 - H_q(\theta - \sqrt{\theta(\theta - \delta)}), \delta \leq \theta$$

A proof of this bound can be found in [12]. When $q = 2$ this can be improved even further (see [15]).

In the opposite direction, the Varshamov-Gilbert bound gives

$$\alpha_q(\delta) \geq 1 - H_q(\delta), \delta \leq \theta.$$

There is a gap between the known upper and lower bounds and it is still unknown what happens there.

Exercises:

1.Let C be an $[n, d]$code over $\mathbf{F}_q$ with $d \geq 2$. Show that,if C contains $(1, 1, \ldots, 1$, then $w(C) \leq n\theta$.

2. If n is odd and $C = \{(0, \ldots, 0), (1, \ldots, 1)\} \subset \mathbf{F}_2^n$ show that C attains the Hamming bound.

3. Compute $A_q(n, w)$ for $(q, n, w) \in \{(2, 6, 1), (2, 7, 1), (2, 7, 2), (3, 5, 1), (3, 5, 2)\}$.