Coding theory, basic notions and notation

Definition. Let Q be an alphabet, $|Q| = q$. For us, the alphabet will always have some extra structure, it will typically be a finite field, but in any case it will be at least an additive group. (So a special element, the neutral element 0 will be distinguished.) A (block) code C of length n is a subset $C \subseteq Q^n$. The sequences in Q^n are called *words*, the elements of C are called *codewords*.We usually denote $|C|$ by M, and call such a code an $(n, M)_q$-code. If $q = 2$, we omit the index.

In order to measure how many errors a code can correct, we introduce a distance d function on Q^n.

Definition. let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two sequences (words) from Q^n. The *Hamming distance* d_H of x, y is the number of indices in which they differ. More formally,

$$d_H(x, y) = d(x, y) = |\{i : x_i \neq y_i\}|.$$

The *Hamming weight* of a word u is its distance from the all-0 word, so $w(u) = d_H(u, 0)$.

It is easy to see that $d_H = d$ is indeed a metric, which means that it is
(0) $d(x, y) \geq 0$
(1) symmetric, that is $d(x, y) = d(y, x)$
(2) $d(x, y) = 0$ if and only if $x = y$, and finally
(3) the *triangle inequality* holds, that is $d(x, z) \leq d(x, y) + d(y, z)$.
Another important property of the Hamming distance is that it is invariant under translations, that is,

$$d(x + c, y + c) = d(x, y).$$

This implies that $d(x, y) = w(y - x)$.

Using the Hamming-distance we can define the minimum distance of a code.

Definition. The *minimum distance* of a code $C \subseteq Q^n$ is the minimum of the distances between two different codewords. More formally, the minimum distance d is

$$\min_{x \neq y \in C} d(x, y).$$

Let us see intuitively what the minimum distance means for error correction. Assume that a codeword $c \in C$ was sent and x was received. If $d(x, c) = t$, then t errors occured. We only know x and want to find the original c. This is called decoding or error correction. This is uniquely possible, if there is no other codeword c', which is at distance at most t from x. The vectors which are at distance at most t form a ball intuitively. Sometimes, we just want to see whether there was an error or not (so detect the errors and not necessarily correct them). This is the case when the data can easily been resent.

Definition. A code C *corrects* t *errors (or* t*-error correcting)* if for every $x \in Q^n$ there is at most one $c \in C$ with $d(x, c) \leq t$. The *ball of radius* r *and centre* c is $B(c, r) = \{x : d(c, x \leq r\}$. So, the code is t-error correcting when the balls of radius t around codewords are pairwise disjoint. The code C *detects* t *errors* if in case of $d(x, c) \leq t$ and $c \in C$, x cannot be a codeword, that is $x \notin C$. In other words, the ball $B(c, t)$ with $c \in C$ cannot contain another codeword $c' \in C$. We also use the expression t*-error detecting*.

These notions can immediately be related to the minimum distance.

Proposition. *If* C *has minimum distance* d*, then it is* t*-error correcting for* $t \leq [(d-1)/2]$*, and it is* $(d-1)$*-error detecting. In other words, a* t*-error correcting code with minimum distance* d *has* $d \geq 2t+1$*, and a* t*-error detecting code with minimum distance* d *has* $d \geq t+1$*.*

We prove the error-correcting part. Assume that $d \leq 2t$. If there are two codewords c and c' with distance at most $2t$, then they differ in at most $2t$ coordinates. Choose t of these coordinates and modify c at these coordinates to have the same coordinate as c'. This way we obtain an x, which has distance t from c, and distance at most t from c'. If the distance of c and c' is at least $2t + 1$, then the balls of radius t around c and c' are indeed disjoint. If they had a common word x, then by the triangle inequality we would have $d(c, c') \leq d(c, x) + d(x, c') \leq 2t$, a contradiction.

There are other types of errors, for example, the so-called erasures. In this case we know when the errors occurs but the actual value is not known. Intuitively, this is better than a random error, because we do not know where it occurs. So, intuitively, this is „half an error", so $d-1$ such erasure errors can be corrected. Yet another possibility is that burst errors occur. This means that whenever we have an error then a sequence of errors occur (imagine that we scratch the CD disk in a wrong direction).

Let us finally see some easy examples for encoding.

1. Binary repetiton code: $b \mapsto (b, b, \ldots b)$, where we have n coordinates of the image.

This code corrects $[(n-1)/2]$ errors.

2. Parity check bit: $(b_1, \ldots, b_{n-1}) \mapsto (b_1, \ldots, b_{n-1}, b_1 + \ldots + b_{n-1})$.

This does not correct any error, it is used in case of computer part (e.g. floppy disks). This code detects 1 error.

3. a special code: $(b_1, b_2, b_3, b_4) \mapsto (b_1, b_2, b_3, b_4, b_1 + b_2 + b_4, b_1 + b_3 + b_4, b_2 + b_3 + b_4)$

This code detects 2 errors and corrects 1 error. It is somewhat tedious to show that any two codewords differ in at least 3 coordinates (so $d = 3$). We will see that this code is the Hamming code Ham(3).

Definition. Two codes C and C' are *equivalent (in a narrow sense)* if permuting the coordinates maps C to C'.

Actually, also the symbols of the alphabet Q can be permuted (however, we distinguished a symbol 0).

Let us also see two general constructions for codes: puncturing and extension. The first one simply means that we delete a coordinate from all codewords, the second one means that we add a parity check bit to all codewords. Over a non-binary alphabet the parity check bit will be computed so that the sum of the coordinates for codewords in the extended code will always be 0. So

$$\bar{C} = \{(c_1, \dots c_{n-1}, -(c_1 + \dots c_{n-1})) : (c_1, \dots c_{n-1}) \in C\}.$$

Puncturing reduces the length by one, and in general reduces the minimum distance by one. Adding the parity check bit may increase the minimum distance.

1. Linear codes

In case of linear codes $Q = \mathrm{GF}(q)$, hence Q^n is a vector space $V(n, q)$.

Definition. A code $C \subseteq \mathrm{GF}(q)^n$ is *linear* if it is a linear subspace. If the dimension is k, and the minimum distance is d, we will denote such a code as an $[n, k, d]_q$ code. Note that in this case $|C| = M = q^k$.

As we saw it for subspaces, C can be described by a generator matrix G or a parity check matrix H. Recall that G is a $k \times n$, H is an $(n - k) \times n$ matrix and the rows are independent in both cases. We can also recall that codewords are linear combinations of the rows of G, and they are perpendicular to the rows of H. In particular, $GH^T = O$. It is also useful to remember that in case $G = (I_k|A)$, we can have $H = (-A^T|I_{n-k})$.

The examples above are all linear codes, the generator matrix for the binary repetition code is $(1\ 1 \dots 1)$, in case of the parity check bit it is $(I_{n-1}|(1\ 1 \dots 1)^T)$ (meaning that the last column is the all-one vector). In case of parity check codes over $\mathrm{GF}(q)$ we have -1 in the last column everywhere. Finally, in case of the length 7 Hamming code (our third example), we have

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

It is easy to find a parity check matrix of the codes, since the first $k \times k$ part of the generator matrix is the identity in all cases. We can also observe that the repetiton code and the parity check bit are dual codes of each other.

The narrow sense equivalence of linear codes can easily be described using their generator or parity check matrix. The permutation of coordinates can be described by multiplying the matrices by an $n \times n$ permutation matrix P. However, for linear codes equivalence is defined in a wider sense.

Definition. Two linear codes C and C' are equivalent if they can be obtained from each other by two types of transformations: permuting the coordinates and multiplying the coordinates by a non-zero scalar (which can be different for different coordinates). In terms of their generator matrices G and G' define equivalent codes when $G' = BGM$, where B is an invertible $k \times k$ matrix and M is a $n \times n$ monomial matrix, that is a matrix in which every row and column contains precisely one non-zero element (it is similar to a permutation matrix, just in place of the „1"-s there are arbitrary non-zero elements.

It is clear that equivalence in terms of the parity check matrix can be described the same way: $H' = BHM$ with B being invertible and M monomial.

Definition. The minimum weight of a linear code C is $\min_{0 \neq c \in C} w(c)$.

For linear codes, the minimum distance and the minimum weight are the same, because the Hamming distance is translation invariant.

In case of linear codes, we can determine the *minimum weight* using the parity check matrix H.

Theorem. *Let H be a parity check matrix of the (linear) code C. The minimum weight (= minimum distance) of C is d if (and only if) every $d-1$ columns of H are linearly independent but there are d columns of H that are dependent.*

Consider a codeword c of weight w. Let the non-zero coordinates be $i_1, \ldots i_w$. Let us denote the corresponding columns of H by o_{i_j}, $j = 1, \ldots, w$. Then cH^T is the linear combination $c_{i_1} o_{i_1} + \ldots + c_{i_w} o_{i_w}$. As $c \in C$, $cH^T = 0$, so if the weight of c is w, then the columns o_{i_j}, $j = 1, \ldots, w$ are linearly dependent. Conversely, if a set of w columns of H is dependent, then we can find a vector c of weight w (consisting of the coefficients of the non-trivial linear combinations) so that $cH^T = 0$.

We see that the proof of the above theorem is relatively easy but the theorem is very important (that is why we called it a theorem and not just a proposition).