Equivalence of codes

The following concept is useful in working with codes. Codes $C$ and $D$ are called equivalent if $D$ can be obtained from $C$ by a sequence of the following “elementary equivalences”.

1. permutating the positions of the codewords

2. for a fixed position, permuting the symbols in that position with a fixed permutation of the alphabet (different permutations are allowed in different positions).

Example: \{000, 111\} $\sim$ \{100, 011\} $\sim$ \{001, 110\}.

Note that if $C$ and $D$ are equivalent codes, then they have the same $n, M, d$ parameters. (Why?)

\[ A_q(n, d). \]

For fixed $n$ and $q$, for an $(n, M, d)_q$ code, there is a trade-off between $M$ and $d$.

Definition: $A_q(n, d)$ is the largest $M$ such that there is an $(n, M, d)_q$ code.

In other words, $A_q(n, d)$ is the size of the largest $q$-ary $(n, M, d)$-code. We will later show that $A_q(n, d)$ is the largest $M$ such that there is an $(n, M, e)_q$-code with $e \geq d$.

A major coding problem: determine $A_q(n, d)$. An upper bound on $A_q(n, d)$ of the form $A_q(n, d) \leq M$ is usually proved by contradiction. On the other hand, a proof of the lower bound of the form $A_q(n, d) \leq N$ usually involves constructing a particular $q$-ary
(n, N, d)-code. In some cases we can prove matching upper and lower bounds (i.e., \( M = N \)) and deduce the exact value of \( A_q(n, d) \). For many \( q, n \) and \( d \) the exact value of \( A_q(n, d) \) is unknown.

Clearly, \( A_q(n, 1) = q^n \). Every code has minimum distance \( \geq 1 \), so this requirement is vacuous. There are exactly \( q^n \) \( q \)-ary words on length \( n \). If we include all of them in a code \( C \), we will obtain a \( q \)-ary \((n, q^n, 1)\)-code. Of course, such a code has no error-detection or error-correction capacity.

Proposition: (i) \( A_q(n, 1) = q^n \), (ii) \( A_q(n, n) = q \),

Proof: (i) The upper bound, \( A_q(n, 1) \leq q^n \) is obvious, since there are only \( q^n \) \( q \)-ary words on length \( n \) in total.

To prove the lower bound, \( A_q(n, 1) \geq q^n \), consider the code \( C \) which includes every \( q \)-ary words on length \( n \). Clearly \( d(C') = 1 \), and thus \( A_q(n, d) \geq |C| = q^n \), as desired.

(ii) If \( C \) is code of length \( n \), then \( d(C') = n \) if and only if all codewords differ in all positions. In particular, the map from \( C \) to \( F \), defined by

\[
f(x_1 x_2 \ldots x_n) = x_1
\]

is 1-1. Thus, \( A_q(n, n) \leq q \). On the other hand, the \( q \)-ary \( n \)-repetition code contains exactly \( q \) codewords. So, \( A_q(n, n) = q \).

Some other specific values (or ranges) of \( A_q(n, d) \) are given in Table 2.4 in the text. It an unsolved problem for some specific values of \( n \) and \( d \), even when \( q = 2 \). There is a patchwork of upper and lower bounds. We will explore some of them in this course.

We will now discuss two constructions that modify a given code. The first construction is called **puncturing**; I will go over it today. The second construction is called **adding the overall parity check**; we will discuss it next week.
Let $C$ be a code and $1 \leq i \leq n$. The we can obtain a new code $D$ of length $n - 1$ from $C$ by deleting the $i$-bit in every codeword:

$$D = \{(x_1 \ldots x_{i-1}x_{i+1} \ldots x_n) : \overline{x} \in C\}$$

This operation is called puncturing (or shortening).

Example 1: The 2-repetition code is a punctured code of the 3-repetition code.

Example 2: The code

$$D = \{0000, 0110, 1010, 1111\}$$

is obtained by puncturing (or shortening)

$$C = C_3 = \{00000, 01110, 10110, 11011\}$$

in the third position.

The puncturing map is $f(\overline{x}) := x_1 \ldots x_{i-1}x_{i+1} \ldots x_n$.

Proposition: $A_q(n, d) \geq A_q(n - 1, d - 1)$ for any $n \geq 2$.

Proof: Start we a $q$-ary code $C$ of length $n$ and minimal distance $d$ of largest possible size $M = A_q(n, d)$. Our goal is to show that there exists a $q$-ary $(n - 1, M, d - 1)$ code $C'$. If we can do this, we will conclude that $A_q(n - 1, d - 1) \geq M$, as desired.

To construct $C'$, choose two words $\overline{x}$ and $\overline{y}$ at distance $d$. Suppose $\overline{x}$ and $\overline{y}$ differ in position $i$. Puncturing $C$ in position $i$, we obtain a new code $C''$ of length $n - 1$. Since $d(C') \geq 2$, the puncturing map is $1 - 1$. That is, $|C'| = |C| = M$. Clearly $d(C'') \geq d - 1$. To see that equality holds, let $\overline{x}'$ and $\overline{y}' \in C'$ be the words obtained from $\overline{x}$ and $\overline{y}$ by deleting the digit in position $i$. Since $d(\overline{x}, \overline{y}) = d$ and $\overline{x}$, $\overline{y}$ differ in position $i$, we have $d(\overline{x}', \overline{y}') = d - 1$. Thus $d(C') \leq d - 1$, as desired.
Corollary (the Singleton bound; see Theorem 10.17 in the book):

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

We will discuss this further in the next lecture.