Math 342. Notes for Lectures 4. January 16, 2018

Last time we discussed two operations on codes, puncturing and parity check. We used puncturing to prove the Singleton bound. Now we will use the parity check operation to prove the following

**Proposition.**

1. If $n \geq d \geq 1$ and $d$ is odd, then $A_2(n + 1, d + 1) \leq A_2(n, d)$.
2. If $m \geq e \geq 1$ and $e$ is even, then $A_2(m, e) = A_2(m - 1, e - 1)$.

Note that part 1 above is false if $d$ is even, and part 2 if $e$ is odd. For example, $A_2(4, 2) = 8$, but $A_2(5, 3) = 4$.

Proof of the Proposition: Part 1 is Theorem 2.7 in the book. Part 2 is just a restatement of part 1, with $e = d + 1$ and $m = n + 1$. □

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\}.

Fact: if $C$ and $D$ are equivalent codes, then they have the same $n, M, d$ parameters.

The following example is mentioned in the book (Example 2.2 in the notes). I did not work it out in class, but am including it here, filling in some of the details left to the reader on p. 14 in the book.

Proposition: $A_2(5, 3) = 4$. 


Proof of Proposition: A $(5, 4, 3)$ binary code, 
\[ \{00000, 01101, 10110, 11011\}, \]
is exhibited in Example 1.5 in the book. Thus $A_2(5, 3) \geq 4$.

It remains to show that $A_2(5, 3) \leq 4$. Assume the contrary. Then there exists a $(5, 5, 3)$ binary code. Denote it by $C$. We seek a contradiction.

By permuting symbols in each position, we may assume that $0 = 00000 \in C$. Thus, all other codewords must have weight 3, 4 or 5.

If $c = 11111 \in C$, then for any codeword $c' \in C, c' \neq 0$, $d(c, c') \leq 2$, a contradiction. So, $C$ can have words of weight only 3 or 4 (other than $0$).

It cannot have more than one word of weight 4, since the distance between any two such words is 2.

Thus, $C$ must have at least 3 codewords of weight 3: $\bar{x}, \bar{y}, \bar{z}$. We will show that this is impossible.

By permuting the codeword positions we may assume that $\bar{x} = 11100 \in C$.

Claim: $\bar{y}$ and $\bar{x}$ have exactly one 1 in a common position (and similarly, $\bar{z}$ and $\bar{x}$ have exactly one 1 in a common position).

Proof of claim: $\bar{x}$ and $\bar{y}$ must have at least one 1 in a common position, since they both weight = 3.

Suppose they have two 1’s in common positions. By a permutation of the first three positions, we may assume that 
\[ \bar{y} = 011?? \]
where one of the ?’s is a 1 and the other is a 0.

But with either choice, $d(\bar{x}, \bar{y}) = 2$, which is impossible.
And \( \overline{x} \) and \( \overline{y} \) cannot have three 1’s in common positions because they are distinct words.

So, they must have only one 1 in common. This proves the claim. □

So, \( \overline{y} = ???11 \) where exactly one of the ? is a 1.

Similarly, \( \overline{z} = ???11 \) where exactly one of the ? is a 1.

Thus, \( \overline{y} \) and \( \overline{z} \) must have a 0 in a common position. Thus, \( d(\overline{y}, \overline{z}) = 2 \), a contradiction to \( d(C) = 3 \).

This proves the proposition. □

Example: As we showed above, \( A_2(n, d) = A_2(n + 1, d + 1) \) if \( d \) is odd. In particular, \( A_2(6, 4) = A_2(5, 3) = 4 \). To construct a binary (6, 4, 4)-code \( D \), start with the the binary (5, 4, 3)-code

\[
C = \{00000, 01110, 10110, 11011\}
\]

and add a parity check digit to each codeword:

\[
D = \{000000, 011101, 101101, 110110\}.
\]

Binomial coefficients

\[
\binom{n}{i} \]

is the number of unordered selections of \( i \) elements out of a set of \( n \) elements.

**Proposition:** \( \binom{n}{i} = \frac{n!}{i!(n-i)!} = \frac{n(n-1)\ldots(n-i+1)}{i!} \).

**Proof.** First count the number of ordered selections of \( i \) elements from \( \{1, \ldots, n\} \). There are

\( n \) ways to choose the first element, \( n - 1 \) ways to choose the second element, \( \ldots \), \( n - i + 1 \) ways to choose the \( i \)th element. All in all,
we obtain \( n(n-1) \ldots (n-i+1) = \frac{n(n-1) \ldots (n-i+1)}{i!} \) ordered selections.

Now given a selection of \( i \) elements out of \( n \), there are \( i! \) ways to order these \( i \) elements. Thus each unordered selection corresponds to exactly \( i! \) ordered selections. In other words,

\[
\text{Number of unordered selections} = \frac{\text{Number of ordered selections}}{i!},
\]

and the proposition follows.

**Proposition:** Let \( \overline{x} = (x_1, \ldots, x_n) \) be a word in \( F_q^n \). Then the number of words \( \overline{y} = y_i = (y_1, \ldots, y_n) \) such that \( d(\overline{x}, \overline{y}) = r \) is

(a) \( \binom{n}{r} \), if \( q = 2 \), and more generally,

(b) \( (q-1)^r \binom{n}{r} \) for arbitrary \( q \geq 2 \).

**Proof.** (a) \( \overline{y} \) is completely determined by the choice of \( r \) positions where it differs from \( \overline{x} \). If \( i \) is one of those positions, then \( y_i \) is uniquely determined by the requirement that \( y_i \) should be different from \( x_i \): if \( x_i = 1 \), then \( y_i = 0 \) and if \( x_i = 0 \), then \( y_i = 1 \). Thus the number of \( \overline{y} \) at distance \( r \) from \( \overline{x} \) is the number of ways to choose \( r \) positions out of \( n \). This number is, by definition, \( \binom{n}{r} \).

(b) \( \overline{y} \) is completely determined by the choice of \( r \) positions where it differs from \( \overline{x} \) and the \( r \) elements that are used to fill these \( r \) positions. If \( i \) is one of these positions, then \( y_i \) can be any element of \( F_q \), other than \( x_i \). there are \( q-1 \) such choices. Thus the number of \( \overline{y} \) at distance \( r \) from \( \overline{x} \) is \( (q-1)^r \binom{n}{r} \). \qed