Problem sheet 2
2004, Jan. 22

MT361 Error correcting codes

Ex. 1

Each properly published book gets a unique ISBN number (international standard book number). This is a 10-digit codeword. The first digit stands for the country/language, the next few digits for the publisher. Then some digits for a number assigned by the publisher, the very last digit is a checksum. (A large publisher gets a short publisher identification and can thus use more digits for its own books, a small publisher gets a longer publisher identification. This alone leads to interesting questions but we leave these aside.)
For example, the recommended text book by Ray Hill has the number
ISBN 0-19-853804-9
ISBN 0-19-853803-0 (for the paperback edition).
Here the first 0 stands for english, the 19 for Oxford University Press.

Let $x_1x_2\cdots x_{10}$ be the ISBN number (codeword). The check bit x_{10} is chosen such that the whole codeword satisfies $\sum_{i=1}^{10} ix_i \equiv 0 \bmod 11$.

a) Show that $x_{10} = \sum_{i=1}^{9} ix_i \equiv 0 \bmod 11$.

   Note that the last symbol can be any of 11 eleven values. So, one uses in addition to $0, 1, \ldots, 9$ the symbol $X = 10$.

b) Show that this code can be used in the following way: To detect any single error and to detect a double error created by the transposition of two digits (example 152784 ↔ 158724).
   Would this also work, if you use a similar code mod 15 instead of mod 11?

c) Can this method be used to correct one single error?

d) Discuss the advantages of this method for the practical use (to order books in a bookshop etc.).

e) What is the minimum distance of any two ISBN numbers?

f) Consider a different code C_2, where one uses as before 10 digits but does not use a weighted sum, but $\sum_{i=1}^{10} x_i \equiv 0 \bmod 11$.
   What would be the disadvantage, compared with the ISBN code?

Ex. 2

a) Show that a 3-ary $(3, M, 2)$-code must have $M \leq 9$.

b) Show that a 3-ary $(3, 9, 2)$-code exists. (Hint: find three codewords starting with 0, and three codeword starting with 1, and three codewords starting with 2).

Ex. 3
(Not to be handed in!)
Work through this example.
$C = \{(00000, 01101, 10110, 11011)\}$ defines a $(5,4,3)$-code. So, $A_2(5,3) \geq 4$. We want to show that no code with $n = 5, M = 5, d = 3$ exists. An exhaustive search would be possible, with a computer. But the following procedure is much more effective:
Let C be a $(5, M, 3)$-code with $M \geq 4$.
By our discussion on equivalent codes we may assume w.l.o.g. that $00000 \in C$. C can contain at most one codeword with weight 4 or 5, since any two such codewords would have distance at most 2. Also, because of $d = 3$ there cannot be any codeword with just one or two ones, since the distance to 00000 would be at most 2. Since $M \geq 4$, there must be at least 2 codewords containg exactly 3 ones. By rearranging the positions we can assume that one of these is 11100. The other one can have at most one of its three ones in the first three position, (otherwise the distance to 11100 would be ≤ 2. So we can assume w.l.o.g. that the third codeword is 00111.
Now, after some trial and error attempts we find that the only possible fourth codeword is 11011. This proves that $A_2(5,3)$.
This type of argument reduces any exhausting search considerably!
It also proves that there is, up to equivalence, exactly one $(5,4,3)$-code.

Ex. 4
Construct, if possible, binary (n, M, d) codes, with the parameters below. If no such code exists, explain why.

a) (6,2,6)

b) (3,8,1)

c) (4,8,2)

d) (5,3,4)

e) (8,30,3)

Ex. 5
Show that $A_2(8,5) = 4$.

Ex. 6
Show that $A_2(8,4) = 16$.

Ex. 7
(Not to be handed in!)
In Example 2.19 we had considered a non-trivial perfect binary $(7, 16, 3)$-code. Make yourself familiar with this example.

$$\begin{array}{rcccccccc}
\vec{0} & = & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vec{a_1} & = & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
\vec{a_2} & = & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
\vec{a_3} & = & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
\vec{a_4} & = & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
\vec{a_5} & = & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\vec{a_6} & = & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\vec{a_7} & = & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\vec{b_1} & = & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
\vec{b_2} & = & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\vec{b_3} & = & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
\vec{b_4} & = & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
\vec{b_5} & = & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
\vec{b_6} & = & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
\vec{b_7} & = & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
\vec{1} & = & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

When evaluating the minimum distance you would need to compare $16 \times 15/2$ pairs. By the cyclical construction this can be much reduced;
Compare $\vec{0}$ with $\vec{1}$ and $\vec{a_1}, \vec{b_1}$. (3)
Compare $\vec{1}$ with $\vec{a_1}, \vec{b_1}$. (2)
Compare $\vec{a_1}$ with $\vec{a_i}, i = 2, 3, \ldots, 7$. (6)
Compare $\vec{a_1}$ with $\vec{b_i}, i = 1, \ldots, 7$. (7)
Compare $\vec{b_1}$ with $\vec{b_i}, i = 2, 3, \ldots, 7$. (6)
These 24 comparisions suffice, (this number can be further reduced by methods that we learn at a later stage in the course). Note that the minimum distance is $d = 3$. Check that the sphere packing bound is sharp here.

Hand in solutions at the beginning of the lecture on Thursday 29th January.
I've put some books in the restricted loan section of the library. Recommended reading is R. Hill: A First course in coding theory. (001.539 Hil)
An electronic version of the problem sheets is available:
http://www.ma.rhul.ac.uk/∼elsholtz/04mt361/lecture.html