C&O 331: Assignment #1 Due date: January 18 (Wednesday)

---

1. (a) The *weight* $\text{wt}(x)$ of a vector x is defined to be the number of its components that are nonzero. Let x and y be two binary vectors of length n. Let $I(x, y)$ denote the number of coordinate positions in which both x and y have 1s; for example $I(1100011, 1011101) = 2$. Prove that $d(x, y) = \text{wt}(x) + \text{wt}(y) - 2I(x, y)$.

   (b) Prove that a binary $[n, M]$-code C of distance $d = 2t+1$ exists if and only if a binary $[n+1, M]$-code C' of distance $d' = 2t + 2$ exists.

2. Consider the binary code $C = \{c_1 = 0000, c_2 = 1110, c_3 = 0111\}$. Suppose that $P(c_1) = 0.1$, $P(c_2) = 0.2$, and $P(c_3) = 0.7$, where $P(c_i)$ denotes the probability that c_i is sent. Suppose that a binary symmetric channel with symbol error probability p is being used, and $r = 1001$ is received.

   (a) What is the distance of C?

   (b) Suppose that $p = 0.1$. Decode r using IMLD.

   (c) Suppose that $p = 0.1$. Decode r using MED.

   (d) Suppose that $p = 0.4$. Decode r using IMLD.

   (e) Suppose that $p = 0.4$. Decode r using MED.

3. (a) Prove the triangle inequality for Hamming distance: that is, $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in A^n$.

   (b) Let C be a code with distance $d = 2t+2$. Prove that C can correct t errors and simultaneously detect $t + 1$ errors.
   (The decoding strategy is the following: if r is received and there is a unique codeword $c \in C$ such that $d(c, r) \leq t$, then r is decoded to c; if there is no such unique codeword, then r is rejected. You have to prove that the decoder always makes a correct decision if $t + 1$ or fewer errors are introduced per codeword.)

4. If x_1 and x_2 are binary n-tuples, then $x_1 + x_2$ denotes the bitwise modulo 2 sum of x_1 and x_2. For example, $000111 + 011011 = 011100$.

   (a) Let C be a binary $[n, M]$-code with distance d. Let $x \in \{0, 1\}^n$, and let $C+x = \{c+x \ : \ c \in C\}$. Prove that the distance of $C + x$ is also d.

   (b) Construct a binary $[8, 4]$-code with distance 5, or prove that no such code exists.

   (c) Construct a binary $[7, 3]$-code with distance 5, or prove that no such code exists.

Please note that assignments are not weighted equally. Each problem on each assignment is worth 10 marks. The total marks received on assignments will be added together at the end of the course.

You should make an effort to solve all the problems on your own. You are also welcome to collaborate with your colleagues, and to seek assistance from the teaching assistant or the instructor. However, *all solutions must be written up by yourself.* If you do collaborate, please acknowledge your collaborators in the write-up for each problem. *If you obtain a solution with help from a book, solutions from previous offerings of the course, a web site (including Wikipedia), or elsewhere, please acknowledge your source.*

The assignment is due by 5:00pm on January 18. Late assignments will not be accepted except in very special circumstances.
