Math 342
Homework Assignment #1 (Due Thursday, January 21, in class)
All numbered problems are worth the same value.

1. How many errors can each of the following codes correct? How many can each detect?

   (a) $\{000000, 111111, 000111\}$, $q = 2$

   (b) $\{11000, 10101, 01110, 00011\}$, $q = 2$

   (c) $\{012345, 123450, 234501, 345012, 450123\}$, $q = 6$

   a. $d(C) = 3$. So, this code can correct 1 error or detect 2 errors.

   b. $d(C) = 3$. So, again this code can correct 1 error or detect 2 errors.

   c. $d(C) = 6$. So, this code can correct 2 errors or detect 5 errors.

2. Do there exist binary codes with the following (n, M, d) parameters? (for each, say why or why not?) (5,3,5), (5,16,2), (5,17,2)

   (5,3,5): in class, we showed that $A_q(n, n) = q$ and so $A_2(5, 5) = 2$. So, there does not exist a (5,3,5) binary code.

   (5,16,2): Yes. Let C be the set of all binary words of length 5 and even weight. Then $d(C) \geq 2$ because if you flip just one bit in a codeword in C you get a word of odd weights. And $d(C) \leq 2$ by virtue of the codewords $00000, 11000$.

   (5,17,2): No. Since there are only 16 binary words of length 4, for any code consisting of 17 words of length 5, at least two of the codewords would agree in their first 4 positions. Thus, those codewords would have distance 1, contradicting minimum distance = 2.

3. Consider the binary symmetric channel with channel error probability $= p$. Let C be the binary 6-repetition code.

   (a) How many errors can be corrected/detected?

   (b) If C is used as an error-correcting code with incomplete nearest-neighbour decoding, what is the probability that it will mis-correct (i.e., that it will either decode to an incorrect codeword or will declare an error when there was no error)?

   (c) If C is used as an error-detecting code, what is the probability that it will mis-detect (i.e., that it will declare an error when there was no error or declare that there is no error when there was an error)?

   a. Since $d(C) = 6$, the code can correct 2 errors or detect 5 errors.

   b. C will decode to an incorrect codeword when and only when 4 or more errors are made. If 3 errors are made, then it will correctly detect an error (there will be a tie between the two codewords). Since C is a 2-error-correcting code, if at most 2 errors are made, then it will correctly correct the error.

   Thus, the probability of mis-correction is

$$\binom{6}{4}p^4(1-p)^2+\binom{6}{5}p^5(1-p)+\binom{6}{6}p^6 = 15p^4(1-p)^2+6p^5(1-p)+p^6$$

   c. It is impossible for C to declare an error when there is no error, since the received word would be a codeword. It will declare that there is no error when there was an error when and only when all 6 bits are flipped. The probability of this event is p^6.

4. Let C be a binary code of length n to be used over the binary symmetric channel with channel error probability $= p$.

For a codeword $\overline{c} = c_1 \dots c_n \in C$ and a word $\overline{x} = x_1 \dots x_n$, we define the probability that $\overline{x}$ was received, given that $\overline{c}$ was transmitted as

$$P(\overline{x}|\overline{c}) = \prod_{i=1}^{n} u_i$$

where $u_i = p$ if $x_i \neq c_i$ and $u_i = 1 - p$ if $x_i = c_i$.

(a) Show that

$$P(\overline{x}|\overline{c}) = p^d(1-p)^{n-d}$$

where $d = d(\overline{x}, \overline{c})$.

(b) In maximum likelihood decoding, the decoder decodes $\overline{x}$ to the codeword $\overline{c}$ which maximizes $P(\overline{x}|\overline{c})$ (if there is a tie, the decoder chooses an arbitrary such $\overline{c}$).

Show that if $p < 1/2$, then maximum likelihood decoding is the same as complete nearest neighbour decoding.

a. If $d = d(\overline{x}, \overline{c})$, then the number of $1 \leq i \leq n$ such that $u_i = p$ is d. For all other i, $u_i = 1 - p$. Thus,

$$P(\overline{x}|\overline{c}) = p^d(1-p)^{n-d}$$

b. Since $p < 1/2$, $p^e(1-p)^{n-e} < p^d(1-p)^{n-d}$ iff $d < e$. Thus, the maximum of $p^d(1-p)^{n-d}$ occurs when and only when d is minimized. And d is minimized for the codeword $\overline{c}$ that is closest, in Hamming distance, to the received word $\overline{x}$.

5. Let C be the binary 4-repetition code. For each of the received words $\overline{x} = 0000, 1000, 1100, 1110, 1111$, say what each of the following decoders will return (i.e., which codeword it will decode to or if it will declare an error)

(a) An incomplete nearest neighbour decoder which corrects 1 error.

(b) A decoder which detects up to 3 errors.

(c) A hybrid decoder that is guaranteed to correct 1 error and detect 2 errors.

received word	incomplete NND	error detector	hybrid
0000	0000	0000	0000
1000	0000	?	0000
1100	?	?	?
1110	1111	?	1111
1111	1111	1111	1111

Note: the hybrid decoder and incomplete NND give the same results. I didn't realize this beforehand. In fact, in all examples, it turns out that the hybrid decoder for $d(C) = 4$ is either the same or inferior to the incomplete NND. However, the hybrid decoder for $d(C) = 4$ is a special case of a much more general class of hybrid decoders which are quite useful. Stay tuned for HW2.

6. The following are converses of results that we proved in class. Let C be a code with $|C| \geq 2$.

(a) Show that if C is u-error-detecting, then $d(C) \geq u + 1$.

(b) Show that if C is v-error-correcting, then $d(C) \geq 2v + 1$.

Let $\overline{x}$ be the received word.

a. Assume that C is u-error detecting. Let $\overline{c}, \overline{c}' \in C$ with $\overline{c} \neq \overline{c}'$. Suppose that $d(\overline{c}, \overline{c}') \leq u$. We seek a contradiction.

$\overline{c}$ could be the transmitted codeword and at most u channel errors could result in $\overline{c}'$ being the received word. In that case, the

decoder would not detect an error, contrary to the assumption that C is u-error detecting.

Thus, $d(\overline{c}, \overline{c}') \geq u + 1$.

b. Assume that C is v-error-correcting. Suppose that $d(C) \leq 2v$.

Let $\overline{c}, \overline{c}' \in C$ s.t. $\overline{c} \neq \overline{c}'$ and

$$d := d(\overline{c}, \overline{c}') \leq 2v.$$

We seek a contradiction.

Case 1: $d \leq v$.

Then $\overline{c}$ could be transmitted and $\overline{c}'$ could be received, making at most v errors. But the NND would return $\overline{c}'$ (incorrectly). This contradicts that C is v-error-correcting.

Case 2: $v < d \leq 2v$.

Without loss of generality (WLOG) we may assume (WMA) that the d positions where they differ come before the positions where they are the same.

Define $\overline{x}$ by

$$x_i = \left\{ \begin{array}{ll} c'_i & i \leq v \\ c_i & i > v \end{array} \right\}$$

Then $d(\overline{c}', \overline{x}) = d - v \leq v = d(\overline{c}, \overline{x})$.

So, if $\overline{c}$ is transmitted and $\overline{x}$ is received, then v errors have been made, and the NND will *not* return $\overline{c}$ as the decoded codeword (at best, it will report a tie between $\overline{c}$ and $\overline{c}'$). This contradicts that C is v-error-correcting.

7. Show that for all $n \geq 2$, $A_2(n, 2) = 2^{n-1}$ and give an explicit example of a code that achieves $A_2(n, 2)$.

By Corollary 2.8 in the text (a more general version done in class), $A_2(n, 2) = A_2(n-1, 1) = 2^{n-1}$.

More directly:

Let C be the code consisting of all binary words of length n and even weight. Then $d(C) \geq 2$ because if you flip just one bit in a codeword in C you get a word with odd weight. And $d(C) \leq 2$ because for any codeword if you flip exactly 2 bits you get another codeword. So, $d(C) = 2$.

And $|C| = 2^{n-1}$ because the mapping $f : C \to \{0,1\}^{n-1}$ given by $f(x_1 \ldots x_n) = x_1 \ldots x_{n-1}$ is 1-1 and onto: it is 1-1 because if not, then there would be two codewords at distance 1; it is onto because for any binary word of length $n-1$, adding an overall parity check results in a word of length n with even weight.

Thus, $A_2(n, 2) \geq 2^{n-1}$.

Let C is a binary code of length n and $d(C) = 2$. If $|C| > 2^{n-1}$ codewords, then two distinct codewords must agree in their first $n-1$ positions and so $d(C) = 1$, a contradiction.

Thus, $A_2(n, 2) \leq 2^{n-1}$ and so $A_2(n, 2) = 2^{n-1}$.

Note: the similarity with arguments used in problem 2 of this HW.

8. Show that for all $n \geq 4$, $A_2(n, n-1) = 2$ and give an explicit example of a code that achieves $A_2(n, n-1)$. What are $A_2(3, 2)$ and $A_2(2, 1)$?

   Let C be a binary code of length n and minimum distance $n-1$. By equivalence, we may assume that $\overline{0}$, the all zeros word, is in the code. Then any other codeword must have weight $n-1$ or n. If there were one codeword $\overline{c}$ of weight $n-1$ and one codeword

$\overline{c}'$ of weight n, then $d(\overline{c}, \overline{c}') = 1 < 3 \leq n - 1$, a contradiction. If there were two distinct codewords $\overline{c}, \overline{c}'$ of weight $n - 1$, then $d(\overline{c}, \overline{c}') = 2 < 3 \leq n - 1$, again a contradiction. Thus, there can be at most two codewords in the code, and the binary repetition code of length n achieves this.

For $n = 2$, we have $n - 1 = 1$ and the code consisting of all 4 binary words of length 2 achieves $A_2(2, 1) = 4$. For $n = 3$, $A_2(n, n - 1) = A_2(3, 2) = A_2(2, 1) = 4$ and is achieved by the four words of length $= 3$ and even weight.