# CSC 310, Spring 2004 — Assignment #3

Due at **start** of tutorial on March 26. Worth 6% of the course grade.

*Note that this assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own.*

**Question 1 (60 marks):** This question concerns a $Z$ Channel, in which channel inputs of 0 are always received correctly, but channel inputs of 1 are received as 0 with probability $f$.

Consider the following three codes for use with the fourth extension of this channel:

$$\begin{aligned} \mathcal{C}_1 &= \{0001,\ 0010,\ 0100,\ 1000\} \\ \mathcal{C}_2 &= \{0101,\ 0110,\ 1001,\ 1010\} \\ \mathcal{C}_3 &= \{1110,\ 1101,\ 1011,\ 0111\} \end{aligned}$$

Since all these codes have four codewords, they could be used to send a block of two message bits.

a) For each of these three codes, derive the maximum likelihood decoder (ie, the decoder which, given the four received bits, finds which codeword was most likely to have been transmitted, assuming that all codewords are sent with equal probability).

You should show your derivation of the decoders, and make clear exactly what the decoding is for all possible received vectors, either by a table or by some more compact explanation. If the decoding is not unique (ie, there are two or more equally good decodings for some received vector), you should make clear what all the possible maximum likelihood decodings are. It is conceivable that the maximum likelihood decoders might depend on the value of $f$ (assumed to be in $(0, 1)$), in which case you should explain the dependence.

*For each of the four codewords, we can figure out the probability of each of the possible received vectors. From these, we can figure out the maximum likelihood decoding for each received vector.*

*Here are the probabilities of the possible received vectors given that each of the four codewords of $\mathcal{C}_1$ was transmitted:*

| | Received vector: | | | | |
|---|---|---|---|---|---|
| | 0001 | 0010 | 0100 | 1000 | 0000 |
| Codeword: 0001 | $1-f$ | 0 | 0 | 0 | $f$ |
| 0010 | 0 | $1-f$ | 0 | 0 | $f$ |
| 0100 | 0 | 0 | $1-f$ | 0 | $f$ |
| 1000 | 0 | 0 | 0 | $1-f$ | $f$ |

*From this, we can conclude that a maximum likelihood decoder will chose the codeword equal to the received vector for received vectors of 0001, 0010, 0100, and 1000, and will choose a codeword arbitrarily if the received vector is 0000. For simplicity in part (b) below, let's assume that the decoder chooses one of the four codewords randomly, with equal probabilities, when 0000 is received. (The error rate would be the same for any other choice when 0000 is received.)*

*Here are the probabilities of the possible received vectors given that each of the four codewords of $\mathcal{C}_2$ was transmitted:*

| | Received vector: | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0101 | 0110 | 1001 | 1010 | 0001 | 0010 | 0100 | 1000 | 0000 |
| Codeword: 0101 | $(1-f)^2$ | 0 | 0 | 0 | $f(1-f)$ | 0 | $f(1-f)$ | 0 | $f^2$ |
| 0110 | 0 | $(1-f)^2$ | 0 | 0 | 0 | $f(1-f)$ | $f(1-f)$ | 0 | $f^2$ |
| 1001 | 0 | 0 | $(1-f)^2$ | 0 | $f(1-f)$ | 0 | 0 | $f(1-f)$ | $f^2$ |
| 1010 | 0 | 0 | 0 | $(1-f)^2$ | 0 | $f(1-f)$ | 0 | $f(1-f)$ | $f^2$ |

*From this, we can conclude that a maximum likelihood decoder will chose the codeword equal to the received vector for received vectors of 0101, 0110, 1001, and 1010. For received vectors 0001, 0010, 0100, and 1000, the decoder will choose arbitrarily one of the two codewords compatible with that received vector. For the received vector 0000, the decoder will choose arbitrarily from all four codewords.*

*Similarly, one can conclude that the maximum likelihood decoder for $C_3$ will decode each possible received vector to a codeword as follows:*

| *Received vector* | *Codeword (or set of codewords)* |
|---|---|
| 0000 | 1110 1101 1011 0111 |
| 0001 | 1101 1011 0111 |
| 0010 | 1110 1011 0111 |
| 0100 | 1110 1101 0111 |
| 1000 | 1110 1101 1011 |
| 0011 | 1011 0111 |
| 0101 | 1101 0111 |
| 1001 | 1101 1011 |
| 0110 | 1110 0111 |
| 1010 | 1110 1011 |
| 1100 | 1110 1101 |
| 1110 | 1110 |
| 1101 | 1101 |
| 1011 | 1011 |
| 0111 | 0111 |

b) For each of the three codes, find the probability that the maximum likelihood decoding is erroneous (ie, produces a codeword other than the one actually sent), assuming that the four codewords are equally likely to be sent. Note that when the decoder can't tell for sure which codeword was sent, it can still guess, and will sometime guess right. The decoder error probabilities may depend on the value of $f$.

*This problem can be solved in at least two ways — by considering each possible transmitted codeword, or by considering each possible received vector. The first way is perhaps easier, since we know that the codewords are all equally likely, whereas the possible received vectors are not equally likely.*

*For code $C_1$, a codeword will be received without error with probability $1-f$, in which case it will be decoded correctly. Otherwise (with probability $f$) the codeword will be received as 0000, in which case, if the decoder chooses one of the four codewords randomly, there will be a 1/4 chance that the decoder chooses the codeword that was actually transmitted. The overall probability of correct decoding is therefore $(1-f) + (1/4)f = 1 - (3/4)f$, so the probability of error is $(3/4)f$.*

*For code $C_2$, a transmitted codeword will be received with no errors, with one error, or with two errors. The probabilities of these situations are $(1-f)^2$, $2f(1-f)$, and $f^2$, respectively. The probabilities of the decoder guessing the right codeword in these situations are 1, 1/2, and 1/4, respectively. The total probability of correct decoding is therefore $(1-f)^2 + (1/2)2f(1-f) + (1/4)f^2 = 1 - f + (1/4)f^2$, so the probability of error is $f - (1/4)f^2$.*

*Similary, for code $C_3$, the probability of correct decoding is*

$$(1-f)^3 + (1/2)3f(1-f)^2 + (1/3)3f^2(1-f) + (1/4)f^3 \;=\; 1 - (3/2)f + f^2 - (1/4)f^3$$

*so the probability of error is $(3/2)f - f^2 + (1/4)f^3$.*

c) Discuss whether or not the results you found for part (b), and in particular, which code is better, are what you would expect in view of the results on mutual information and capacity for the $Z$ channel that were presented in lectures and are found in the textbook.

*Comparing the error probabilities for $C_1$, $C_2$, and $C_3$, one can see that the $C_1$ has the lowest error probability for all $f \in (0,1)$. We saw in lecture 9a that the capacity of the Z channel is reached with an input distribution in which 0 is more probable than 1. In a qualitative way, it therefore makes sense that $C_1$, which has codewords in which 0.75 of the bits are 0, might be better than $C_2$ and $C_3$, in which the fractions of 0 bits are 0.5 and 0.25. However, we can't expect an exact match with the input distribution that maximizes the mutual information, since this depends on $f$, and may have a value that can't be exactly achieved using codewords that are four bits long. Also, it may be that none of these three codes is the best possible code.*

*If we used longer codewords, we would expect that the best code would have codewords in which the fraction of 0 bits was closer to the distribution that maximizes mutual information.*

**Question 2 (40 marks):** Suppose that $\mathcal{C}_H$ is a code for the $H$th extension of a binary channel, and the $\mathcal{C}_V$ is a code for the $V$th extension of a binary channel. The *product* of $H$ and $V$ is a code for the $HV$th extension of the channel. A codeword of this product code can be visualized as a two-dimensional array with $V$ rows and $H$ columns, in which all the rows are codewords of $\mathcal{C}_H$ and all the columns are codewords of $\mathcal{C}_V$. The product code consists of all such $V$ by $H$ arrays.

a) Give an example of two non-empty codes $\mathcal{C}_H$ and $\mathcal{C}_V$ for which the product code is empty (ie, for which it is not possible to construct a $V$ by $H$ array with the rows being codewords of $H$ and the columns being codewords of $V$).

*One example (there are many): Let $\mathcal{C}_H$ be the set of all vectors of length three with exactly one 1 bit — ie, $\mathcal{C}_H = \{001, 010, 100\}$. Let $\mathcal{C}_V$ be the set of all vectors of length three with exactly one 0 bit — ie, $\mathcal{C}_V = \{110, 101, 011\}$. Clearly, in any 3 by 3 array in which the rows are codewords of $\mathcal{C}_H$, 1/3 of the bits will be 1. But in any 3 by 3 array in which the columns are codewords of $\mathcal{C}_V$, 2/3 of the bits will be 1. These are not compatible conditions, so the product code will be empty.*

b) Give a one-sentence proof that if $\mathcal{C}_H$ and $\mathcal{C}_V$ are linear codes, their product code is non-empty.

*The product of two linear codes will always have the array containing all zeros as a codeword, since the rows and columns of an all-zero array will be vectors of all zeros, which are codewords in any linear code.*

c) Suppose that $C_V$ is the repetition code of length three, and that $C_H$ is a single-parity-check code of length two or more. Give an time-efficient decoding algorithm for this product code that is guaranteed to correctly decode whenever the received block has no more than two errors. Discuss whether or not it is possible to correctly decode some or all received vectors that have three errors, giving examples to show that it is sometimes possible, and/or sometimes not possible to guarantee correct decoding with three errors.

*An algorithm guaranteed to correct any two errors: First, look at all the columns of the received array of bits, and for any column in which the bits are not all the same, flip the bit that is different from the other two. Second, compute the three parity checks on the rows. If none of these parity checks are satisfied, and at least one of the columns was changed in the first stage of decoding, flip all the bits in the first column that was changed in the first stage.*

*The minimum distance of $C_V$ is 3 and the minimum distance of $C_H$ is 2, so the minimum distance of the product code is $3 \times 2 = 6$. This is not enough to guarantee correction of any three errors (for which we would need a minimum distance of 7). Nevertheless, it is possible to correct some patterns of three errors, including any pattern in which the errors occur in different columns. An example of a pattern of three errors which cannot be corrected (except by luck) is when the three errors are all in the same column. In this case, all the columns of the received array will be codewords of $C_V$, but none of the rows will be codewords of $C_H$. We could conclude that something is wrong, but we would have no way of knowing which column contains the errors.*