CSC 310, Fall 2011 — Solutions to Theory Assignment #3

Question 1 (35 marks): Consider a channel for which the input alphabet and output alphabet are both $\{0, 1, 2\}$, and for which the channel transition probabilities are given by $Q_{0|0} = Q_{2|2} = 1$, $Q_{0|1} = Q_{2|1} = 1/4$, and $Q_{1|1} = 1/2$.

a) Find the mutual information between the channel input and output if the input probabilities are $p_0 = 1/2$, $p_1 = 0$, and $p_2 = 1/2$.

Let the channel input be X and the channel output be Y. The mutual information can be written as $I(X;Y) = H(X) - H(X|Y)$. With the input distribution above, $H(X) = 1$ bit. $H(X|Y)$ is zero, since if symbol 1 is never sent, symbol 1 will never be received, and if symbol 0 or 2 is received, the symbol that was sent must be the same as what was received. The mutual information is therefore 1 bit.

b) Find the mutual information between the channel input and output if the input probabilities are $p_0 = p_1 = p_2 = 1/3$.

We can write the mutual information as $I(X;Y) = H(Y) - H(Y|X)$. We can find the output probabilities as $q_0 = p_0 + (1/4)\,p_1 = 5/12$, $q_1 = (1/2)\,p_1 = 1/6$, and $q_2 = p_2 + (1/4)\,p_1 = 5/12$. Hence

$$H(Y) = (5/12)\log_2(12/5) + (1/6)\log_2(6) + (5/12)\log_2(12/5) = 1.483$$

$H(Y|X) = (1/3)\,H(Y|X=0) + (1/3)\,H(Y|X=1) + (1/3)\,H(Y|X=2)$. *Since both $H(Y|X=0)$ and $H(Y|X=2)$ are zero, we get that*

$$\begin{aligned} H(Y|X) &= (1/3)\,H(Y|X=1) \\ &= (1/3)\left[(1/4)\,\log_2(4) + (1/2)\,\log_2(2) + (1/4)\,\log_2(4)\right] \\ &= (1/3)\,(3/2) = 0.5 \end{aligned}$$

The mutual information is therefore $1.483 - 0.5 = 0.983$ bits.

c) Find the capacity of this channel, and an input distribution that achieves this capacity. Once you have reduced this to a one-dimensional optimization problem, you may use a numerical method to find the solution, written in any language you choose. Hand in the command or program that you use, which may be quite short if you're using something like Maple. Giving the capacity in bits to three decimal places is sufficient, and a brute force numerical search is acceptable.

We need to maximize $I(X;Y)$ with respect to p_0, p_1, and p_2, which must be positive and sum to one.

First, we can show that $I(X;Y)$ will be maximized with an input distribution in which $p_0 = p_2$. To see this, we fix p_1 and consider changing p_0, and hence $p_2 = 1 - p_1 - p_0$. The output probabilities will be $q_0 = p_0 + (1/4)\,p_1$, $q_1 = (1/2)\,p_1$, and $q_2 = p_2 + (1/4)\,p_1$. We then write $I(X;Y) = H(Y) - H(Y|X)$, and note that $H(Y|X)$ depends only on p_1, not on p_0 or p_2. Furthermore, for a given value of p_1, and hence q_1, $H(Y)$ is maximized

when q_0 and q_2 are equal, which will happen when p_0 and p_2 are equal. So if we fix p_1, the maximum value of $I(X;Y)$ will be when p_0 and p_2 are both $(1-p_1)\,/\,2$.

Maximizing $I(X;Y)$ now reduces to a one-dimensional optimization problem. As for part (b), we see that $H(Y|X) = (3/2)p_1$, and when $p_0 = p_2$, we can write $q_0 = q_2 = (2-p_1)/4$, so that

$$H(Y) \;=\; -(1-p_1/2)\log_2((2-p_1)/4) \;-\; (p_1/2)\log_2(p_1/2)$$

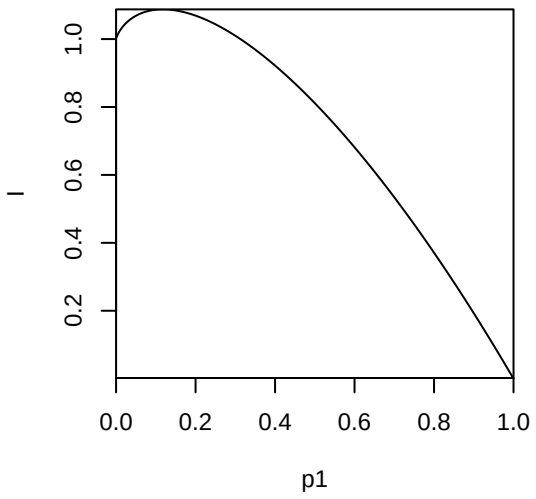
The mutual information when $p_0 = p_2$ is therefore

$$I(X;Y) \;=\; -(1-p_1/2)\log_2((2-p_1)/4) \;-\; (p_1/2)\log_2(p_1/2) \;-\; (3/2)p_1$$

The following R commands plot this function, and find its maximum and the value of p_1 where this maximum is acheived:

```
> p1 <- seq(0.001,0.999,by=0.001)
> I <- -(1-p1/2)*log((2-p1)/4,2) - (p1/2)*log(p1/2,2) - (3/2)*p1
> plot(p1,I,type="l",xaxs="i",yaxs="i",xlim=c(0,1))
> max(I)
[1] 1.087462
> p1[which.max(I)]
[1] 0.118
```

The maximum of the mutual information, and hence the channel capacity, is 1.087 bits, acheived when $p_1 = 0.118$ and $p_0 = p_1 = 0.441$. Here is the plot produced above:



Question 2 (30 marks): Suppose that we send a two-bit message by sending five bits through a Binary Symmetric Channel with error probability 1/3, using the $[5, 2]$ linear code from the lectures, in which the codewords are 00000, 00111, 11001, and 11110. Suppose that these four codewords are equally likely to be sent, and suppose that the decoder decodes by maximum likelihood. What is the probability that the decoder will decode to the wrong codeword? Does this error probability depend on any arbitrary choices made by the decoder?

In the lecture slides, a syndrome decoding table was found for this code, as follows (with an entry for a syndrome of zero added):

z	**n**
000	00000
001	00001
010	00010
011	00100
100	01000
101	10000
110	10100
111	01100

For each possible syndrome, this table gives an error pattern of minimal weight that produces that syndrome. If we decode using this syndrome table, we will correct all (and only) the error patterns that appear in it. Since errors are independent and occur with probability 1/3, the probability of correct decoding is

$$(2/3)^5 \ + \ 5(2/3)^4(1/3) \ + \ 2(2/3)^3(1/3)^2 \ = \ 0.5267$$

The probability of erroneous decoding is therefore $1 - 0.5267 \ = \ 0.4733$.

The last two entries in the syndrome table are not unique — there are other patterns of two errors that also produce those syndromes. A different maximum likeihood decoder could therefore decode differently when one of these syndromes occurs. However, any error patterns chosen for these two syndromes will have two errors, since the entry in the syndrom table has to be an error pattern with mimimal weight. So the calculation of the probability of decoding error above would be no different.

Question 3 (30 marks): Suppose that we use a linear code on a binary alphabet (ie, a linear subspace of Z_2^N) to encode messages that are sent over a Binary Symmetric Channel. Suppose also that decoding is done by maximum likelihood (and in case of ties, a codeword is selected uniformly at random from among all those with the maximum likelihood). Prove that the probability that the decoded message is not what was sent is the same regardless of which message was sent.

One way to prove this is to consider the implementation of maximum likelihood using the syndrome decoding table. You would need to expand the table to hold all minimal-weight error patterns that produce a given syndrome, so the decoder could choose among them randomly, and then argue that this is a correct implementation of maximum likelihood decoding, and that the probability of error using this implementation doesn't depend on the codeword sent.

There is a much more direct proof, however. The probability of decoding error will certainly not depend on which codeword was sent if the stronger statement holds that for every error pattern, the probability that decoding is successful does not depend on which codeword was sent. We can prove that stronger statement.

Consider some error pattern $\mathbf{n}$ *(a vector in* Z_2^N*), and two codewords,* $\mathbf{u}_1$ *and* $\mathbf{u}_2$*. We want to show that if error pattern* $\mathbf{n}$ *occurs, the probability of correct decoding when* $\mathbf{u}_1$ *is sent is the same as when* $\mathbf{u}_2$ *is sent.*

When $\mathbf{u}_1$ is sent, the decoder will receive $\mathbf{v}_1 = \mathbf{u}_1 + \mathbf{n}$. There are three possibilities for how this will be decoded:

1) *Decoding will be incorrect because there is anaother codeword, $\mathbf{u}'_1$, that is closer in Hamming distance to $\mathbf{v}_1$ than $\mathbf{u}_1$ (that is, $\mathbf{v}_1 - \mathbf{u}'_1$ has lower weight than $\mathbf{v}_1 - \mathbf{u}_1 = \mathbf{n}$).*

2) *Decoding will be correct because the Hamming distance from $\mathbf{v}_1$ to $\mathbf{u}_1$ is less than the Hamming distance from $\mathbf{v}_1$ to any other codeword, $\mathbf{u}'_1$.*

3) *Decoding will be correct with probability $1/h$, with h an integer greater than one, because there are $h-1$ other codewords, $\mathbf{u}'_1$, $\mathbf{u}''_1$, ..., at the same distance from $\mathbf{v}_1$ as $\mathbf{u}_1$, and there are no codewords that are closer to $\mathbf{v}_1$ than $\mathbf{u}_1$.*

If instead $\mathbf{u}_2$ is sent, with the error pattern again being $\mathbf{n}$, the decoder will receive $\mathbf{v}_2 = \mathbf{u}_2 + \mathbf{n}$. Now, define $\mathbf{d} = \mathbf{u}_2 - \mathbf{u}_1$, and note that since the code is linear, $\mathbf{d}$ is a codeword. We have that $\mathbf{v}_2 = \mathbf{v}_1 + \mathbf{d}$. Now consider how $\mathbf{v}_2$ is decoded in each of the three situations listed above:

1) *In this case, $\mathbf{v}_1$ is decoded incorrectly, as $\mathbf{u}'_1$, and $\mathbf{v}_2$ will also be decoded incorrectly. To see this, let $\mathbf{u}'_2 = \mathbf{u}'_1 + \mathbf{d}$, which will be a codeword since $\mathbf{u}'_1$ and $\mathbf{d}$ are codewords. The distance from $\mathbf{v}_2$ to $\mathbf{u}'_2$ will be the same as the distance from $\mathbf{v}_1$ to $\mathbf{u}'_1$, since $\mathbf{v}_2 - \mathbf{u}'_2 = (\mathbf{u}_1 + \mathbf{d} + \mathbf{n}) - (\mathbf{u}'_1 + \mathbf{d}) = \mathbf{v}_1 - \mathbf{u}'_1$. Since this distance is less than the weight of $\mathbf{n}$, $\mathbf{v}_2$ will be incorrectly decoded to $\mathbf{u}'_2$ (or to some other codeword that is also closer to $\mathbf{v}_2$ than $\mathbf{u}_2$ is).*

2) *In this case, $\mathbf{v}_1$ is decoded correctly, since all the incorrect codewords are further from $\mathbf{v}_1$ than $\mathbf{u}_1$ is, and $\mathbf{v}_2$ will also be decoded correctly. To see this, suppose that some codeword, $\mathbf{u}'_2$, other than $\mathbf{u}_2$ is not further from $\mathbf{v}_2$ than $\mathbf{u}_2$ is (ie, the weight of $\mathbf{v}_2 - \mathbf{u}'_2$ is not greater than the weight of $\mathbf{n}$). Then the codeword $\mathbf{u}'_1 = \mathbf{u}'_2 - \mathbf{d}$ (which is not equal to $\mathbf{u}_1$) would not be further from $\mathbf{v}_1$ than $\mathbf{u}_1$ is, since $\mathbf{v}_1 - \mathbf{u}'_1 = (\mathbf{v}_2 - \mathbf{d}) - (\mathbf{u}'_2 - \mathbf{d}) = \mathbf{v}_2 - \mathbf{u}'_2$ does not have weight greater than the weight of $\mathbf{n}$. But this contradicts the assumption that all codewords other than $\mathbf{u}_1$ are further from $\mathbf{v}_1$ than $\mathbf{u}_1$. So all the codewords other than $\mathbf{u}_2$ must be further from $\mathbf{v}_2$ than $\mathbf{u}_2$, and hence $\mathbf{v}_2$ is decoded correctly.*

3) *In this case, $\mathbf{v}_1$ has probability $1/h$ of being decoded correctly, and the same is true for $\mathbf{v}_2$. Arguing as above, one can see that there are no codewords closer to $\mathbf{v}_2$ than $\mathbf{u}_2$, and that the set of other codewords at equal distance is $\mathbf{u}'_1 + \mathbf{d}$, $\mathbf{u}''_1 + \mathbf{d}$, ..., which number $h-1$. So the probability of $\mathbf{v}_2$ being decoded correctly is also $1/h$.*

So in all cases the probability of correctly decoding when $\mathbf{u}_2$ is sent and the error pattern is $\mathbf{n}$ is the same as the probability of correctly decoding when $\mathbf{u}_1$ is sent and the error pattern is again $\mathbf{n}$, which implies what we wish to prove.