Rutgers University
The State University of New Jersey
College of Engineering
Department of Electrical and Computer Engineering

330:542 | Information Theory and Coding | Spring 1997

Quiz 2 Solutions

1. *(50 points)* **Source Coding**

   (a) *(25 points)* A computer generates a number X according to a known distribution $p_X(x)$ where $x \in \{1, 2, ...100\}$. The player asks a question, "Is $X = i$?" and is told either "yes," "you're too high," or "you're too low." The player continues for a total of six questions. If correct (i.e., the answer "yes" is received at some point in the sequence) the player receives $v(X)$ dollars. How should the player proceed to maximize earnings over many plays of the game? What is the value of the expected earnings? Note that $v()$ is an arbitrary real-valued function.

   This was a thought problem. The first thing to realize is that you either win or you lose on each trial. That is, the prize is chosen (a number) and you either guess it or you don't in the six alotted trials. So, let's look at this carefully. In the first trial you guess a number. If it's the right number you stop. If not then it's either too high and you guess another number or it's to low and you guess a third number. Proceeding this way we see that that we can identify at most 63 distinct numbers in the six guesses we have.

   Now the question becomes which 63 numbers should we try to guess. Well, that's easy. We want to use the ones that maximize the expected earnings. So, we calculate $w(x) = v(x)p(x)$ for each x and take the 63 largest $w(x)$.

   A little more analytically, our expected earnings are

   $$E = \sum_{x \in W} p(x)v(x)$$

   To maximize E, it's pretty obvious we want to populate the set W with the largest $p(x)v(x)$.

   Now finally, we need a guessing method. Well, that's easy too. We just take our 63 numbers and arrange our guesses to perform a binary search. That is, our first guess x_1 will be the "middle" number (31 numbers higher and 31 numbers lower than x_1). If too high, then we know it's in the lower 31 and if too low we know it's in the upper. Proceeding with this binary search, if the number chosen is in W, we will find it.

   (b) *(25 points)* Consider a variation of the game where now questions can be of any form desired and only "yes" or "no" answers are given. The game stops when the number is determined. That is, the game stops when by the sequence of answers the number can be uniquely determined. This does not necessarily correspond to a single yes answer. Questions cost one dollar each. What is the best strategy for the player given $v()$? What is the expected value of the earnings?

   The expected earnings are

   $$E = \sum_x p(x)(v(x) - l(x))$$

   where $l(x)$ is the number of guesses we use to find x. Looking closely at E we see that it's really two sums,

   $$E = \sum_x p(x)v(x) - \sum_x p(x)l(x)$$

The first sum is fixed no matter what we do since we only stop when the number is guessed. We only have control over the second term. Well, that second sum is the negative of the mean expected codeword length (number of yes/no questions is identical to the number of bits needed to code the answer!). So we just us the Huffman procedure to minimize the average codeword length and thereby maximize our earnings.

2. *(50 points)* **Discrete Channel Capacity**

   (a) *(25 points)* The "Z" channel has inputs zero and one and outputs zero and one. The transition probabilities are $p_{0|0} = 1$, $p_{1|0} = 0$ and $p_{0|1} = p_{1|1} = 1/2$. Make a sketch for reference. Find the capacity of the "Z" channel and the maximizing input probability distribution.

   Capacity is a max over $p(x)$ of $I(X:Y)$. $I(X:Y) = H(Y) - H(Y|X)$ (we could also use the other identity, by the way). Let the probability that $X = 0$ be p so $p(y=0) = (p+1)/2$. Note then that $H(Y|x=0) = 0$ and that $H(Y|x=1) = 1$ (assuming $\log_2$). So in total we have

   $$I(X:Y) = -(\frac{p+1}{2}\log\frac{p+1}{2} + \frac{1-p}{2}\log\frac{1-p}{2}) - (1-p)$$

   Taking the derivative and setting to zero we have

   $$-1/2 - \frac{1}{2}\log\frac{p+1}{2} + 1/2 + \frac{1}{2}\log\frac{1-p}{2} + 1 = 0$$

   or

   $$\log\frac{p+1}{1-p} = 2$$

   which means

   $$\frac{p+1}{1-p} = 4$$

   so that $p = 3/5$. This makes intuitive sense since we should use the 0 input more frequently than the 1 input since it's more reliable. The capacity is then $C = 0.322$ bits (but you did not need to calculate this exactly).

   (b) *(25 points)* Suppose we generate $M = 2^{nR}$ binary codewords of length n for the "Z" channel by tossing a fair coin Mn times; i.e., each n tosses constitute a codeword. This channel input distribution (equiprobable) will not achieve capacity. However, there is some maximum rate R at which we can guarantee asymptotically reliable transmission – that is, the average probability $P_e^{(n)} \to 0$ in n. Find this maximum rate R.

   By specifying the input distribution, we've taken away the only degree of freedom in maximizing the mutual information. Thus, we are stuck with what we're given. That is, the maximum achievable rate is the mutual information between X and Y which is completely specified by $p = 1/2$. Evaluating the mutual information at $p = 1/2$ gives approximately $R = 0.31$. So it looks like we don't pay THAT big a penalty for using the wrong input distribution.

3. *(50 points)* **Differential Entropy and Channel Capacity**

   (a) *(25 points)* A non-negative random variable X is exponentially distributed as $f_Z(z) = a\mathbf{e}^{-az}$ with $a > 0$. Calculate the entropy of Z. Remember that the mean of an exponential distribution with parameter a is $1/a$.

   Straight calculation:

   $$h(X) = -\int_0^\infty a\mathbf{e}^{-az}\log a\mathbf{e}^{-az}dz = 1 - \log a$$

(b) *(25 points)* A memoryless channel has two possible inputs $X \in \{\alpha, \beta\}$ where $\alpha < \beta$ are real constants. The channel output is $Y = X + Z$ where Z is a non-negative exponentially distributed random variable with parameter a. If an average power constraint $E[X^2] \leq P$ is imposed on X, what is the capacity of this channel? What values of α, β and probability mass function $p_X(x)$ achieve capacity.

HINT: This might be tricky so think broadly; i.e., there's more than one correct approach.

One way to approach this problem is from first principles. $I(X:Y) = h(Y) - h(Y|X)$, but $h(Y|X)$ is independent of the distribution on X because $h(Y|X = \alpha) = h(Y|x = \beta)$ by the shifting theorem. Thus, we seek to maximize $h(Y)$.

Let $p = \text{Prob}\{X = \alpha\}$. Then

$$f_Y(y) = pae^{-a(y-\alpha)}u(y-\alpha) + (1-p)ae^{-a(y-\beta)}u(y-\beta)$$

where $u()$ is the unit step function. We first show that $h(Y)$ is monotone increasing in $\beta - \alpha = d$. We have after some manipulation

$$h(Y) = \int_0^d pae^{-ay} \log \frac{e^{ay}}{pa} dy - (1 - p(1-e^{-ad})) \int_0^\infty ae^{-ay} \log[ae^{-ay}(1 - p(1-e^{-ad}))]dy$$

After evaluating the second integral we have

$$h(Y) = \int_0^d pae^{-ay} \log(pae^{-ay})dy + (1 - p(1-e^{-ad})[1 - \log a - \log(1 - p(1-e^{-ad}))]$$

We want to see if $h(Y)$ is monotone increasing in d. To do this we take the derivative with respect to d to obtain

$$\frac{\partial h(Y)}{\partial d} = ape^{-ad}[ad - \log p + \log(1 - p(1-e^{-ad}))]$$

We now need to determine whether the derivative is always greater than zero (monotone increasing in d. We can discard the factor ape^{-ad} since it is always positive for nonzero a and finite d. So, to prove positivity we have

$$ad - \log p + \log(1 - p(1-e^{-ad})) = ad - \log \frac{p}{1 - p(1-e^{-ad})}$$

or

$$e^{ad} \stackrel{?}{>} \frac{p}{1 - p(1-e^{-ad})}$$

$$(1-p)e^{ad} + p \stackrel{?}{>} p$$

which is true $\forall d \geq 0$. So for any given value of p we make d as large as possible to maximize $h(Y)$.

But how large can we make d? For fixed p we have our power constraint of

$$p\alpha^2 + (1-p)\beta^2 = p\alpha^2 + (1-p)(\alpha^2 + 2d\alpha + d^2) = \alpha^2 + 2dp\alpha + pd^2 < P$$

The left side is monotone increasing in d for fixed α and p. Since we desire a maximum d we seek an α which minimizes the left side for any given d. This minimization yields $\alpha = -dp$ so that we have

$$pd^2 - d^2p^2 = d^2p(1-p) < P$$

The absolute maximum d in terms of p and P is then

$$d^2 = \frac{P}{p(1-p)}$$

The only thing left to do is find the right value of p. However, this is difficult. If you got this far (nobody did :), you received full credit.

Another approach is to look at a single channel use. There are four possibilities: α is sent and β is decided or β is sent and α is decided (errors) or we correctly decide α or β. We look at the likelihood ratios letting p be the probability that α was sent. If

$$\frac{pf(Y|x=\alpha)}{(1-p)f(Y|x=\beta)} > 1$$

we decide in favor of $x = \alpha$ and in favor of $x = \beta$ if the converse is true. It is pretty easy to see that we always decide in favor of α when $y < \beta$. Also, as long as $(1-p) > pe^{-a(\beta-\alpha)}$ we decide in favor of β when $y \geq \beta$. Otherwise we ALWAYS decide in favor of α and this leads to a channel with ZERO CAPACITY. So let's assume we choose p such that $\frac{1-p}{p} > e^{-a(\beta-\alpha)}$.

When looked at this way, the channel looks like a Z channel with crossover probability ϵ instead of 1/2. If β is sent, we always correctly decode it. If α is sent we make an error whose magnitude is given by

$$\epsilon = p\int_{\beta-\alpha}^{\infty} ae^{-ay}dy = pe^{-a(\beta-\alpha)} = pe^{-ad}$$

which is monotone decreasing in d.

We then do the same business with constraining α and β in terms of p and the power constraint. Finding p is still tricky however owing to the dependency between p and ϵ through the power constraint; that is,

$$d^2 = \frac{P}{p(1-p)}$$

so

$$\epsilon = pe^{-a\sqrt{\frac{P}{p(1-p)}}}$$

We'd then need to calculate the mutual information between input and our processed output (the guess $\hat{X}$). This is

$$I(X:\hat{X}) = H(\hat{X}) - H(\hat{X}|X) = \mathcal{H}_B(p(1-\epsilon)) - p\mathcal{H}_B(\epsilon)$$

and minimizing in p is painful :)