EE6340 - Information Theory

Problem Set 5 Solution

1. Since all instantaneous codes are uniquely decodable, $L_2 \leq L_1$.Any set of codeword lengths that achieve $\min(L_2)$ will satisfy Kraft inequality and hence we can construct an instantaneous code with the same codeword lengths, and hence the same L.Thus $L_1 \leq L_2$. Both conditions together imply $L_1 = L_2$.

2. Instantaneous codes are prefix codes, i.e, no codeword is a prefix of any other codeword.Let $n_{max} = \max\{n_1, n_2, ..., n_q\}$. There are $D^{n_{max}}$ sequences of length n_{max}. Of these, $D^{n_{max}-n_i}$ sequences start with the i^{th} codeword.Because of the prefix condition, no two sequences can start with the same codeword. Hence the total number of sequences that start with some codeword is $\sum_{i=1}^{q} D^{n_{max}-n_i} = D^{n_{max}} \sum_{i=1}^{q} D^{-n_i} < D^{n_{max}}$. Hence there are sequences that do not start with any codeword. These and all longer sequences with these length n_{max} codewords as prefixes cannot be decoded.This situation can be best visualised using a tree.

   Alternatively, we can map codewords onto dyadic intervals on the real line corresponding to the real umbers whose decimal expansions start with the codewords. Since the length of the interval for a codeword of length n_i is D^{-n_i} and $\sum D^{-n_i} < 1$,there exist some intervals not used by any codeword. The sequences in these intervals do not begin with any codeword and hence cannot be decoded.

3. A possible solution for optimal codes for each state can be

Next state	S_1	S_2	S_3
Code C_1	0	10	11
Code C_2	10	0	11
Code C_3	-	0	1

   Average message length of the next symbol conditioned on the previous state using the given coding scheme is

$$\mathbb{E}(L|C_1) = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{4}(2) = 1.5$$
$$\mathbb{E}(L|C_2) = \frac{1}{4}(2) + \frac{1}{2}(1) + \frac{1}{4}(2) = 1.5$$
$$\mathbb{E}(L|C_3) = 0(1) + \frac{1}{2}(1) + \frac{1}{2}(1) = 1$$

   Note that this code assignment achieves the conditional entropy lower bound.

   To find the unconditional average, we have to find the stationary distribution on the states. Let μ be the stationary distribution. Then solving $\mu = \mu P$

$$\implies \mu = \mu \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 \\ 0 & 1/2 & 1/2 \end{bmatrix} \implies \mu = \begin{bmatrix} 2/9 & 4/9 & 1/3 \end{bmatrix}$$

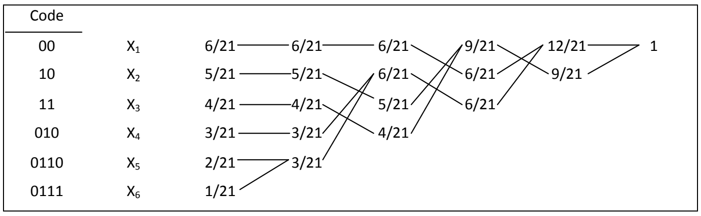
   Unconditional average no. of bits per source symbol is

$$EL = \sum_{i=1}^{3} \mu_i \mathbb{E}(L|C_i) = 4/3$$

Entropy rate of the Markov chain $H = H(X_2|X_1) = \sum_{i=1}^{3} \mu_i H(X_2|X_1 = S_i) = 4/3$.

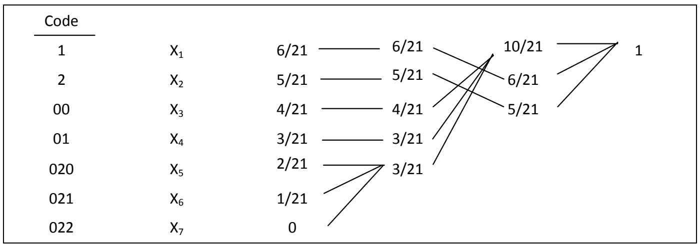
We observe that the unconditional average no. of bits per source symbol = Entropy rate of the Markov chain because the expected length of each code C_i = Entropy of state after $S_i(H(X_2|X_1 = S_i))$, so compression is maximal.

4. Binary Huffman code



$\mathbb{E}$(length)=51/21=2.43 bits.

Ternary Huffman code



$\mathbb{E}$(length)=34/21=1.62 bits.

5. a) With 6 questions, the player cannot cover more than 63 values. This can be seen by induction. With 1 question, he can cover 1 value, with 2, he can cover 1 value with first question and depending on this answer, there are 2 possible values of X that can be asked in the next question. By extending this argument, we can see that he can ask 63 different questions of the form "Is X=i" with 6 questions.

   Thus if a player wants to maximise his expected return, he should choose the 63 outcomes which have the highest values of $p(x)v(x)$ and play to isolate these values.

   First question should be "Is X=i" where i is the median of these 63 values. After isolating one half using the first question, the second question must be "Is X=j" where j is the median of the half remaining after the first question. The maximum expected winnings will be sum of the 63 chosen $p(x)v(x)$.

   b) If arbitrary questions are allowed, the game reduces to 20 questions to determine the object. Returns = $\sum_x p(x)(v(x) - l(x))$, where $l(x)$=no.of questions required to determine the object. Maximising the expected return is equivalent to minimising the expected no.of questions, and thus the optimal strategy is to construct a Huffman code for the source and use that to construct a question strategy.

$$\sum_x p(x)v(x) - H - 1 \leq \text{Expected return} \leq \sum_x p(x)v(x) - H$$

   c) A computer wishing to minimise the return to the player will want to minimise $\sum_x p(x)v(x) - H(x)$ over choices of $p(x)$. Note that this is only a lower bound to the expected winning of

the player. Although the expected winnings of the player may be larger, we will assume that the computer wants to minimise the lower bound.
Let $J(p) = \sum p_i v_i + \sum p_i \log p_i + \lambda \sum p_i$
Differentiating and setting to 0, $v_i + \log p_i + 1 + \lambda = 0$
After normalising to ensure p is a pmf, $p_i = \frac{2^{-v_i}}{\sum_j 2^{-v_j}}$
Now let $r_i = \frac{2^{-v_i}}{\sum_j 2^{-v_j}}$

$$\sum_i p_i v_i + \sum_i p_i \log p_i = \sum_i p_i \log p_i + \sum_i p_i \log 2^{-v_i}$$
$$= \sum_i p_i \log p_i - \sum_i p_i \log r_i - \log(\sum_i 2^{-v_j}) = D(p||r) - \log(\sum_i 2^{-v_j})$$

Thus return is minimised by choosing $p_i = r_i$. This is the distribution that the computer must choose.

6. a) We need to minimise $C = \sum_i p_i c_i l_i$ such that $2^{-l_i} \leq 1$. We will assume equality in the constraint and let $r_i = 2^{-l_i}$ and let $Q = \sum_i p_i c_i$. Let $q_i = (p_i c_i)/Q$. Then q also forms a probability distribution and we can write C as

$$C = \sum_i p_i c_i l_i = Q \sum_i q_i \log \frac{1}{r_i}$$
$$= Q(\sum_i q_i \log \frac{q_i}{r_i} - \sum_i q_i \log q_i)$$
$$= Q(D(q||r) + H(q))$$

We can minimise C by choosing $r = q$ or $l_i{}^* = -\log \frac{p_i c_i}{\sum_j p_j c_j}$. Here we ignore any integer constraints on $l_i{}^*$. The minimum cost C^* for this assignment $= QH(q)$.

b) If we use q instead of p for the Huffman procedure, we obtain a code minimising the expected cost.

c) Now we account for integer constraints, let $l_i = \lceil -\log q_i \rceil$.
Then, $-\log q_i \leq l_i \leq -\log q_i + 1$
Multiplying by $p_i c_i$ and summing over i, we get
$C^* \leq C_{huffman} \leq C^* + Q$

7. a) Since $l_i = \lceil \log \frac{1}{p_i} \rceil$, we have

$$\log \frac{1}{p_i} \leq l_i \leq \log \frac{1}{p_i} + 1$$
$$\implies H(X) \leq L = \sum_i p_i l_i < H(X) + 1 \quad (1)$$

The difficult part is to prove that the code is a prefix code. By the choice of l_i, we have $2^{-l_i} \leq p_i \leq 2^{-(l_i-1)}$ Thus, F_j, $j > i$ differs from F_i by atleast 2^{-l_i} and will therefore differ from F_i in atleast one place in the first l_i places. Thus no codeword is a prefix for any other codeword.

Symbol	Probability	F_i in decimal	F_i in binary	l_i	Code
1	0.5	0.0	0.0	1	0
2	0.25	0.5	0.10	2	10
3	0.125	0.75	0.110	3	110
4	0.125	0.875	0.111	3	111

b) The Shannon code in this case achieves the entropy bound(1.75 bits) and is optimal.