Fundamentals of Information Theory

Solution 3

Problem 1 *(10 points)*

(a) *Find the entropy rate of the two-state Markov chain with transition matrix*

$$P = \begin{bmatrix} 1 - p_{01} & p_{01} \\ p_{10} & 1 - p_{10} \end{bmatrix}.$$

(b) *What values of p_{01}, p_{01} maximize the rate of part (a)?*

(c) *Find the entropy rate of the two-state Markov chain with transition matrix*

$$P = \begin{bmatrix} 1 - p & p \\ 1 & 0 \end{bmatrix}.$$

(d) *Find the maximum value of the entropy rate of the Markov chain of part (c). We expect that the maximizing value of p should be less than $\frac{1}{2}$, since the 0 state permits more information to be generated than the 1 state.*

Solution: Entropy rates of Markov chains.

(a) The stationary distribution is easily calculated. At the steady state,

$$\mu_0 = (1 - p_{01})\mu_0 + p_{10}\mu_1,$$
$$\mu_1 = p_{01}\mu_0 + (1 - p_{10})\mu_1,$$
$$\mu_0 + \mu_1 = 1.$$

Thus,

$$\mu_0 = \frac{p_{10}}{p_{01} + p_{10}}, \mu_1 = \frac{p_{01}}{p_{01} + p_{10}}.$$

Therefore the entropy rate is

$$H(X_2|X_1) = \mu_0 H(p_{01}) + \mu_1 H(p_{10}) = \frac{p_{10}H(p_{01}) + p_{01}H(p_{10})}{p_{01} + p_{10}}.$$

(b) The entropy rate is at most 1 bit because the process has only two states. This rate can be achieved if (and only if) $p_{01} = p_{10} = \frac{1}{2}$, in which case the process is actually i.i.d. with $Pr(X_i = 0) = Pr(X_i = 1) = 1/2$.

(c) As a special case of the general two-state Markov chain, the entropy rate is

$$H(X_2|X_1) = \mu_0 H(p) + \mu_1 H(1) = \frac{H(p)}{p + 1}.$$

(d) By straightforward calculus, we find that the maximum value of $H(X)$ of part (c) occurs for $p = (3 - \sqrt{5})/2 = 0.382$. The maximum value is

$$H(p) = H(1 - p) = H(\frac{\sqrt{5} - 1}{2}) = 0.694 bits.$$

Note that $\frac{\sqrt{5}-1}{2} = 0.618$ is (the reciprocal of) the Golden Ratio.

Problem 2 *(10 points)*

(a) *Are Shannon codes compact codes? Why?*

(b) *Explain why you can obtain a prefix code according to the shannon coding algorithm?*

Solution:

(a) No, Shannon codes are not compact codes. It can achieve the minimum average code length only when the source symbols are uniformly distributed. The upper bound of optimal code lengths doesn't necessarily result in a good code. The codeword for infrequent symbol is usually longer in the Shannon code.

(b)
- To obtain an instantaneousness code, no codeword should be a prefix of others.
- Consider codeword $z_1, z_2 \dots z_l$.
- It represents not a point but the interval

$$\left[0.z_1 z_2 \dots z_l, 0.z_1 z_2 \dots z_l + \frac{1}{2^l}\right)$$

.
- The code is instantaneous (prefix-free), if and only if the corresponding intervals are disjoint.
- The step corresponds to x_i lies in the range from $p_a(x_{i-1})$ to $p_a(x_i)$, which has a height of $p_a(x_i) - p_a(x_{i-1}) = p(x_{i-1})$.
- We have $l_i = \lceil \log\left(\frac{1}{p(x_i)}\right) \rceil$.
- The interval corresponding to any coderword has length 2^{l_i}, which satisfies $2^{-l_i} \leq p(x_i) \leq p(x_{i-1})$.
- The lower end of the interval is $\lfloor p_a(x_i) \rfloor_{l_i}$, and the upper end of the interval is $\lfloor p_a(x_i) \rfloor_{l_i} + 2^{l_1}$.
- Prove the intervals corresponding to different codewords are disjoint, i.e., $\lfloor p_a(x_{i+1}) \rfloor_{l_{i+1}} \geq p_a(x_i) \rfloor_{l_i} + 2^{l_1}$.

Problem 3 *(10 points) Consider the following method for generating a code for a random variable X which takes on m values $\{1, 2, \dots, m\}$ with probabilities $p_1, p_2, \dots p_m$. Assume that the probabilities are ordered so that $p_1 \geqslant p_2 \geqslant \dots \geqslant p_m$. Define*

$$F_i = \sum_{k=1}^{i-1} p_k,$$

the sum of the probabilities of all symbols less than i. Then the codeword for i is the number $F_i \in [0, 1]$ rounded off to l_i bits, where $l_i = \left\lceil \log \frac{1}{p_i} \right\rceil$.

(a) *Show that the code constructed by this process is prefix-free and the average length satisfies*

$$H(X) \leq L < H(X) + 1$$

(b) *Construct the code for the probability distribution* $(0.5, 0.25, 0.125, 0.125)$.

Solution:

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

$$\log \frac{1}{p_i} \leqslant l_i < \log \frac{1}{p_i} + 1,$$

which implies that

$$H(X) \leqslant L = \sum p_i l_i < H(X) + 1.$$

The difficult part is to prove that the code is a prefix code. By the choice of l_i, we have

$$2^{-l_i} \leqslant p_i < 2^{-(l_i - 1)}.$$

Thus F_j, $j > i$ differs from F_i by at least 2^{-i_i}, and will therefore differ from F_i at least one place in the first l_i bits of the binary expansion of F_i. Thus the codeword for F_j, $j > i$,which has length $l_j \geqslant l_i$, differs from the codeword for F_i at least once in the first l_i places. Thus no codeword is a prefix of any other codeword.

(b) We build the following table.

Symbol	Probability	F_i in decimal	F_i in binary	l_i	Codeword
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

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

Problem 4 *(10 points) Consider the random variable*

$$X = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ 0.49 & 0.26 & 0.12 & 0.04 & 0.04 & 0.03 & 0.02 \end{pmatrix}.$$

(a) Find a binary Huffman code for X.

(b) Find the expected codelength for this encoding.

(c) Find a ternary Huffman code for X.

Solution:

(a) The Huffman tree for this distribution is

Codeword								
1	x_1	0.49	0.49	0.49	0.49	0.49	0.51	1
00	x_2	0.26	0.26	0.26	0.26	0.26	0.49	
011	x_3	0.12	0.12	0.12	0.13			
01000	x_4	0.04	0.05	0.08	0.12			
01001	x_5	0.04	0.04	0.05				
01010	x_6	0.03	0.04					
01011	x_7	0.02						

(b) The expected length of the codewords for the binary Huffman code is 2.02 bits. ($H(X) = 2.01$bits)

(c) The ternary Huffman tree is

Codeword					
0	x_1	0.49	0.49	0.49	1.0
1	x_2	0.26	0.26	0.26	
20	x_3	0.12	0.12	0.25	
22	x_4	0.04	0.09		
210	x_5	0.04	0.04		
211	x_6	0.03			
212	x_7	0.02			

This code has an expected length 1.34 ternary symbols. ($H_3(X) = 1.27$ ternary symbols).

Problem 5 *(10 points) Bad wine. One is given 6 bottles of wine. It is known that precisely one bottle has gone bad (tastes terrible). From inspection of the bottles it is determined that the probability p_i that the ith bottle is bad is given by $(p_1, p_2, \ldots, p_6) = (\frac{8}{23}, \frac{6}{23}, \frac{4}{23}, \frac{2}{23}, \frac{2}{23}, \frac{1}{23})$. Tasting will determine the bad wine. Suppose you taste the wines one at a time. Choose the order of tasting to minimize the expected number of tastings required to determine the bad bottle. Remember, if the first 5 wines pass the test you don't have to taste the last.*

(a) What is the expected number of tastings required?

(b) Which bottle should be tasted first?

Now you get smart. For the first sample, you mix some of the wines in a fresh glass and sample the mixture. You proceed, mixing and tasting, stopping when the bad bottle has been determined.

(c) What is the minimum expected number of tastings required to determine the bad wine?

(d) What mixture should be tasted first?

Solution:

(a) If we taste one bottle at a time, to minimize the expected number of tastings the order of tasting should be from the most likely wine to be bad to the least. The expected number of tastings required is

$$\begin{aligned}\sum_{i=1}^{6} p_i l_i &= 1 \times \frac{8}{23} + 2 \times \frac{6}{23} + 3 \times \frac{4}{23} + 4 \times \frac{2}{23} + 5 \times \frac{2}{23} + 5 \times \frac{1}{23} \\ &= \frac{55}{23} \\ &= 2.39\end{aligned}$$

(b) The first bottle to be tasted should be the one with probability $\frac{8}{23}$.

(c) The idea is to use Huffman coding. With Huffman coding, we get codeword lengths as $(2, 2, 2, 3, 4, 4)$. The expected number of tastings required is

$$\begin{aligned}\sum_{i=1}^{6} p_i l_i &= 2 \times \frac{8}{23} + 2 \times \frac{6}{23} + 2 \times \frac{4}{23} + 3 \times \frac{2}{23} + 4 \times \frac{2}{23} + 4 \times \frac{1}{23} \\ &= \frac{54}{23} \\ &= 2.35\end{aligned}$$

(d) The mixture of the first and second bottles should be tasted first.