CS258 Information Theory Homework 5

Zhou Litao 518030910407 F1803016

March 27, 2020

Exercise 1 (Slackness in the Kraft inequality) An instantaneous code has word lengths $l_1, l_2, \ldots, l_m$, which satisfy the strict inequality $\sum_{i=1}^{m} D^{-l_i} < 1$ The code alphabet is $\mathcal{D} = \{0, 1, 2, \ldots, D-1\}$. Show that there exist arbitrarily long sequences of code symbols in D^* which cannot be decoded into sequences of codewords.

Proof. W.l.o.g., we assume $l_1 \le l_2 \le \ldots \le l_m$. Since the coding method is instantaneous, for any coding with the length of l_i, there exists no other coding that begins with the l_i corresponding coding. That is to say, if there exists a coding with l_i length, then $D^{l_m - l_i}$ codewords of the D^{l_m} codewords will be decodable. Adding all these decodable words up we have

$$\sum_{i=1}^{m} D^{l_m - l_i} = D^{l_m} \sum_{i=1}^{m} D^{-l_i} < D^{l_m} \tag{1}$$

,which implies that there exists some prefix words in D^{l_m} that are undecodable. Any arbitrary word that begins with such prefix will be unable to decode into sequence of codewords. $\square$

Exercise 2 (Fix-free codes) A code is a fix-free code if it is both a prefix code and a suffix code. Let $l_1, l_2, \ldots, l_m$ be m positive integers. Prove that if $\sum_{k=1}^{m} 2^{-l_k} \le \frac{1}{2}$ then there exists a binary fix-free code with codeword length $l_1, l_2, \ldots, l_m$

Proof. W.l.o.g., we assume $l_1 \le l_2 \le \ldots \le l_m$. We prove by induction on m.

When $m = 1$, the conclusion is trivial.

We assume that when $m = k-1$, for any increasingly ordered positive integers $l_1, l_2, \ldots, l_{k-1}$, if $\sum_{i=1}^{k-1} 2^{-l_i} \le \frac{1}{2}$, then there exists a binary fix-free code with codeword length $l_1, l_2, \ldots, l_{k-1}$.

Now let $m = k$. For every particular codeword, in terms of prefix codes, it will occupy the space of $2^{l_m - l_i}$ in the tree, while in terms of suffix codes, it wll also occpy the space of $2^{l_m - l_i}$ in the 2^{l_m} nodes. These two sets may overlap, but they will surely be less than $2 \times 2^{l_m - l_i}$. Hence for all codewords, we have

$$2 \sum_{i=1}^{k} 2^{l_m - l_i} \le 2^{l_m} \tag{2}$$

It follows that $\sum_{i=1}^{k} 2^{-l_i} \le \frac{1}{2}$. By removing l_k we have $\sum_{i=1}^{k-1} 2^{-l_i} < \frac{1}{2}$. By the induction hypothesis, we know that for $l_1, \ldots, l_{k-1}$ lengths of codewords, they can form a set of fix-free codes. Furthermore, the strict less relation tells us that there still remains space for the codeword l_k. Hence, the conclusion follows. $\square$

Exercise 3 ($\frac{3}{4}$ fix-free codes) Prove that when

$$\sum_{k=1}^{m} 2^{-l_k} \le \frac{3}{4}$$

the conclusion above holds.

Proof. The proof here is not complete. We try to solve the problem from two perspectives. The proof idea is based on this paper[1].

[1] R. Ahlswede and B. Balkenhol and L. Khachatrian. Some properties of Fix-Free Codes

There exists no upper bounds greater than $\frac{3}{4}$. For any $\frac{3}{4}+\epsilon > \frac{3}{4}$, we choose k such that $2^{-k} < \epsilon$. We construct a list of codelengths with 1 and $2^{k-2}+1$ ks. Then we have

$$\sum_{i=1}^{N} 2^{-l_i} = \frac{1}{2} + 2^{-k}(2^{k-2}+1) = \frac{3}{4} + \epsilon \tag{3}$$

Our choice of codeword lengths satisfies our assumption, however, with the first codeword with length of 1 as a prefix and suffix, there are at most 2^{k-2} words of length k, which implies that our choice is invalid, contradiction.

The conclusion with $\frac{3}{4}$ holds under some restrictions. We suppose that for all code lengths, either $l_i = l_{i+1}$ or $2l_i \le l_{i+1}$. We prove that under this restriction the conclusion will hold.

W.l.o.g., we assume that $l_1 \le l_2 \le \ldots \le l_m$. We prove by induction on m. The base case of $m = 1$ is trivial.

Assume that for any $n \le m-1$, with $\sum_{i=1}^{n} 2^{-l_i} \le \frac{3}{4}$ we can construct n different codeword lengths. We prove that this holds for the case of m.

Let m' be the largest index i with $l_i < l_{m'}$. By induction hypothesis we can construct a fix-free code C' with the lengths $l_1, \ldots, l_{m'}$. Note for every particular word with length l_i, it will occupy at most $2 \times 2^{l_m - l_i}$ nodes due to the prefix and suffix rule. However, with our restriction, $2^{l_m - 2l_i}$ nodes will be returned, since they will not actually be used as codewords. Therefore, at the l_N level, at most $2\sum_{i=1}^{m'} 2^{l_m - l_i} - \sum_{i=1}^{m'} 2^{l_{m'} - 2l_i}$ nodes will be occupied.[2]

To ensure that the remaining $l_{m'+1}, \ldots, l_m$ can be added to the orginal code system we should have

$$2\sum_{i=1}^{m} 2^{l_m - l_i} - \sum_{i=1}^{m} 2^{l_m - 2l_i} \le 2^{l_m} - (m - m') \tag{4}$$

Writing $K = m - m'$ and $\alpha = \sum_{i=1}^{m'} 2^{-l_i}$. (4) can be written as

$$2\alpha - \alpha^2 \le 1 - \frac{K}{2^{l_m}} \tag{5}$$

With abbreviation $\beta = \sum_{i=1}^{m} 2^{-l_i} = \alpha + \frac{K}{2^{l_m}}$ and $\delta = \frac{K}{2^{l_m}}$ we get the equivalent inequality

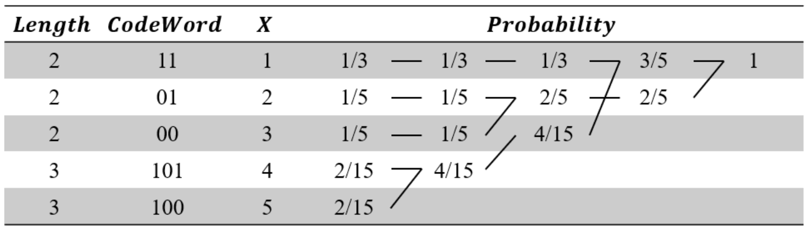
$$\beta \le 1 + \delta - \sqrt{\delta} \tag{6}$$

Note $\delta \in (0,1)$, the right side has the minimal value of $\frac{3}{4}$ at $\delta = \frac{1}{4}$. Thus for any $\sum_{i=1}^{m} 2^{-l_i} \le \frac{3}{4}$, the conclusion holds.

□

Exercise 4 (More Huffman codes) Find the binary Huffman code for the source with probabilities $\left(\frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{2}{15}, \frac{2}{15}\right)$. Argue that this code is also optimal for the source with probabilities $\left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right)$

Figure 1: Huffman Code for $\left(\frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{2}{15}, \frac{2}{15}\right)$.

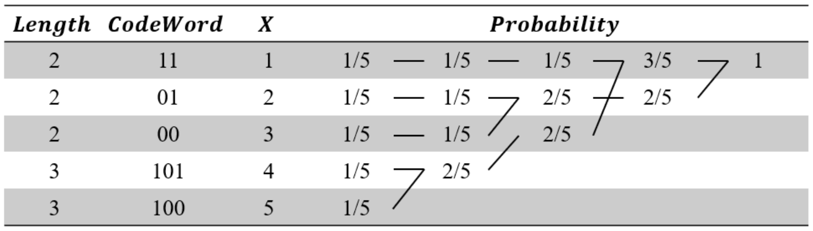


Length	*CodeWord*	*X*	*Probability*				
2	11	1	1/3	1/3	1/3	3/5	1
2	01	2	1/5	1/5	2/5	2/5	
2	00	3	1/5	1/5	4/15		
3	101	4	2/15	4/15			
3	100	5	2/15				

Solution. The huffman code of $\left(\frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{2}{15}, \frac{2}{15}\right)$. is found as Figure 1 shows.

[2]The paper gives a more detailed formula, but I can't fully understand how the last component formula is derived. Fortunately leaving the that component out will not affect the proof here.

Figure 2: Huffman Code for $\left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right)$



Length	*CodeWord*	*X*	*Probability*				
2	11	1	1/5	1/5	1/5	3/5	1
2	01	2	1/5	1/5	2/5	2/5	
2	00	3	1/5	1/5	2/5		
3	101	4	1/5	2/5			
3	100	5	1/5				

We see that for $\left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right)$, we can follow an identical process to construct the huffman code, as Figure 2 shows. Since huffman code is always optimal, the conclusion holds. □

Exercise 5 (Bad codes) Which of these codes cannot be Huffman codes for any probability assignment?

1. {0,10,11}
2. {00,01,10,110}
3. {01,10}

Solution.

1. {0,10,11} can be the huffman code for distribution $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}$.
2. {00,01,10,110} can be shortened by {00,01,10,11}. It cannot be a huffman code.
3. {01,10} can be shortened by {0,1}. It cannot be a huffman code.

□

Exercise 6 (Huffman 20 questions) Consider a set of n objects. Let $X_i = 1$ or 0 accordingly as the i th object is good or defective. Let $X_1, X_2, \ldots, X_n$ be independent with $\Pr\{X_i = 1\} = p_i$; and $p_1 > p_2 > \cdots > p_n > \frac{1}{2}$. We are asked to determine the set of all defective objects. Any yes-no question you can think of is admissible.

1. Give a good lower bound on the minimum average number of questions required.
2. If the longest sequence of questions is required by nature's answers to our questions, what (in words) is the last question we should ask? What two sets are we distinguishing with this question? Assume a compact (minimum average length) sequence of questions.
3. Give an upper bound (within one question) on the minimum average number of questions required.

Solution.

1. The asking process can be modeled into the problem of constructing a compressed code for the sequence $X_1, X_2, \ldots, X_n$. The set of all defective objects can be determined if we determine the codeword we've constructed. The question we are asking is a yes-no question, so the codeword is based on a 2-ray alphabet. We can use entropy of $X_1, X_2, \ldots, X_n$ to give a lower bound on the average codeword length.

$$\begin{aligned} L^* &\geq H_2(X_1, X_2, \ldots, X_n) \\ &= \sum_{i=1}^{n} H(X_i) = \sum_{i=1}^{n} H(p_i) \end{aligned} \tag{7}$$

2. The longest sequence implies that we are distinguishing between the two least cases in this problem, i.e. the case where all objects are good $(\prod_{i=1}^{n}(1-p_i))$ and the case where all objects are good except the one that has the least probability to be defective $(p_n \prod_{i=1}^{n-1}(1-p_i))$. The question will be like "Is X_n defective?".

3. Using the same notion in part 1, the upper bound of the minimum average number of questions (codeword length) will be

$$\begin{aligned} L^* &\le H_2(X_1, X_2, \ldots, X_n) + 1 \\ &= \sum_{i=1}^{n} H(X_i) + 1 = \sum_{i=1}^{n} H(p_i) + 1 \end{aligned} \tag{8}$$

□

Exercise 7 (Simple optimum compression of a Markov source) Consider the three-state Markov process $U_1, U_2, \ldots$ having transition matrix

U_n \ U_{n-1}	S_1	S_2	S_3
S_1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
S_2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
S_3	0	$\frac{1}{2}$	$\frac{1}{2}$

Thus, the probability that S_1 follows S_3 is equal to zero. Design three codes C_1, C_2, C_3 (one for each state 1,2 and 3, each code mapping elements of the set of S_i 's into sequences of 0 's and 1 's, such that this Markov process can be sent with maximal compression by the following scheme:

1. Note the present symbol $X_n = i$
2. Select code C_i
3. Note the next symbol $X_{n+1} = j$ and send the codeword in C_i corresponding to j
4. Repeat for the next symbol.

What is the average message length of the next symbol conditioned on the previous state $X_n = i$ using this coding scheme? What is the unconditional average number of bits per source symbol? Relate this to the entropy rate $H(\mathcal{U})$ of the Markov chain.

Solution. We can design the codes using the Huffman method.

	S_1	S_2	S_3
C_1	1	01	00
C_2	01	1	00
C_3	N/A	0	1

We first calculate the stationery distribution of the Markov Chain.

$$\begin{cases} \mu P = \mu \\ \mu \mathbf{1}^T = 1 \end{cases} \Rightarrow \mu = \left[\frac{2}{9}, \frac{4}{9}, \frac{1}{3}\right] \tag{9}$$

The average length is

$$\sum_{i=1}^{3} \mu_i \sum_{j=1}^{3} L(C_{ij}) = \frac{2}{9} \cdot \frac{3}{2} + \frac{4}{9} \cdot \frac{3}{2} + \frac{1}{3} \cdot 1 = \frac{4}{3} \tag{10}$$

The entropy rate can be calculated as

$$H(\mathcal{U}) = -\sum_{ij} \mu_i P_{ij} \log P_{ij} = \frac{4}{3} \tag{11}$$

They are the same because the optimal code length is its entropy, which can be expressed with the notion of "average" entropy $\frac{1}{n} H(U_1, U_2, \ldots, U_n, \ldots)$, which approximates to $H(\mathcal{U})$. □

Exercise 8 (Shannon codes and Huffman codes) Consider a random variable X that takes on four values with probabilities $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12}\right)$

1. Construct a Huffman code for this random variable.
2. Show that there exist two different sets of optimal lengths for the codewords; namely, show that codeword length assignments (1,2,3,3) and (2,2,2,2) are both optimal.
3. Conclude that there are optimal codes with codeword lengths for some symbols that exceed the Shannon code length $\left\lceil \log \frac{1}{p(x)} \right\rceil$

Solution.

1. A Huffman code is constructed as Figure 3 shows.

Figure 3: Huffman Code for $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12}\right)$

Length	***CodeWord***	***X***	***Probability***			
1	1	1	1/3	1/3	1/3	1
2	01	2	1/3	1/3	2/3	
3	001	3	1/4	1/3		
3	000	4	1/12			

2. Another Huffman code is constructed as Figure 4 shows. They are both optimal for the distribution but with distinct length assignments.

Figure 4: Huffman Code for $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12}\right)$

Length	***CodeWord***	***X***	***Probability***			
2	11	1	1/3	1/3	2/3	1
2	10	2	1/3	1/3	1/3	
2	01	3	1/4	1/3		
2	00	4	1/12			

3. The case of $X = 3$ in Figure 3 serves as an example. It has the length of 3, exceeding the Shannon length $\left\lceil \log \frac{1}{p(x)} \right\rceil = 2$. It can be concluded that in some cases, we can construct optimal codes with codeword lengths for some symbols that exceed the Shannon code length.

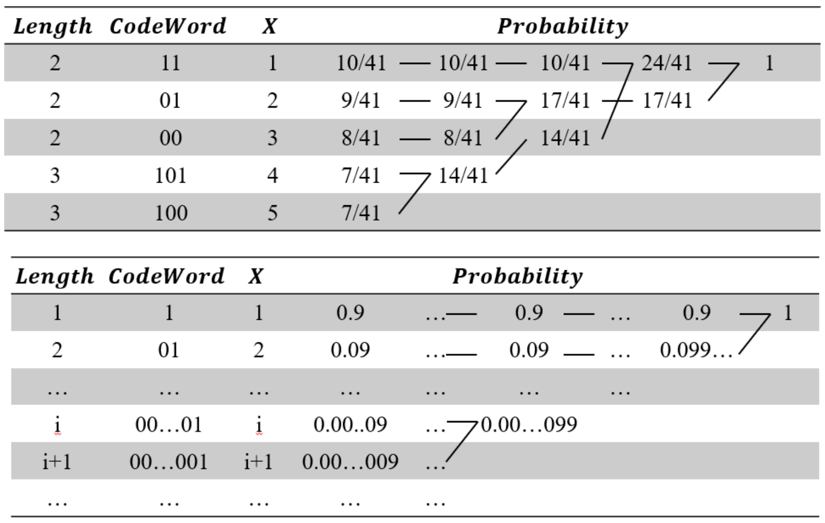
□

Exercise 9 (Data compression) Find an optimal set of binary codeword lengths $l_1, l_2, \ldots$ (minimizing $\sum p_i l_i$) for an instantaneous code for each of the following probability mass functions:

1. $\mathbf{p} = \left(\frac{10}{41}, \frac{9}{41}, \frac{8}{41}, \frac{7}{41}, \frac{7}{41}\right)$
2. $\mathbf{p} = \left(\frac{9}{10}, \left(\frac{9}{10}\right)\left(\frac{1}{10}\right), \left(\frac{9}{10}\right)\left(\frac{1}{10}\right)^2, \left(\frac{9}{10}\right)\left(\frac{1}{10}\right)^3, \ldots\right)$

Solution. The optimal code is given in Figure 5 by the Huffman rule. Note that in problem (2) we have that any probability is greater than the sum of the probabilities less than itself. Hence, we can construct the Huffman code in a monotonous order □

Figure 5: Huffman Code for Exercise 9



Length	*CodeWord*	*X*	*Probability*				
2	11	1	10/41	10/41	10/41	24/41	1
2	01	2	9/41	9/41	17/41	17/41	
2	00	3	8/41	8/41	14/41		
3	101	4	7/41	14/41			
3	100	5	7/41				

Length	*CodeWord*	*X*	*Probability*					
1	1	1	0.9	…	0.9	…	0.9	1
2	01	2	0.09	…	0.09	…	0.099…	
…	…	…	…	…	…	…		
i	00…01	i	0.00..09	…	0.00…099			
i+1	00…001	i+1	0.00…009	…				
…	…	…	…	…				