COMP2610/COMP6261 – Information Theory

Tutorial 9

Zhifeng Tang (zhifeng.tang@anu.edu.au)

Question 1.

Shannon codes and Huffman codes. Consider a random variable X which takes on four values with probabilities $(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12})$.

(a) Construct a Huffman code for this random variable.

(b) 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.

(c) Conclude that there are optimal codes with codeword lengths for some symbols that exceed the Shannon code length $\lceil \log \frac{1}{p(x)} \rceil$.

Solution: *Shannon codes and Huffman codes.*

(a) Applying the Huffman algorithm gives us the following table

Code	Symbol	Probability			
0	1	1/3	1/3	2/3	1
11	2	1/3	1/3	1/3	
101	3	1/4	1/3		
100	4	1/12			

which gives codeword lengths of 1,2,3,3 for the different codewords.

(b) Both set of lengths 1,2,3,3 and 2,2,2,2 satisfy the Kraft inequality, and they both achieve the same expected length (2 bits) for the above distribution. Therefore they are both optimal.

(c) The symbol with probability $1/4$ has an Huffman code of length 3, which is greater than $\lceil \log \frac{1}{p} \rceil$. Thus the Huffman code for a particular symbol may be longer than the Shannon code for that symbol. But on the average, the Huffman code cannot be longer than the Shannon code.

Question 2.

Huffman code. Find the (a) *binary* and (b) *ternary* Huffman codes for the random variable X with probabilities

$$p = (\frac{1}{21}, \frac{2}{21}, \frac{3}{21}, \frac{4}{21}, \frac{5}{21}, \frac{6}{21}) \ .$$

(c) Calculate $L = \sum p_i l_i$ in each case.

Solution: *Huffman code.*

(a) The Huffman tree for this distribution is

Codeword							
00	x_1	6/21	6/21	6/21	9/21	12/21	1
10	x_2	5/21	5/21	6/21	6/21	9/21	
11	x_3	4/21	4/21	5/21	6/21		
010	x_4	3/21	3/21	4/21			
0110	x_5	2/21	3/21				
0111	x_6	1/21					

(b) The ternary Huffman tree is

Codeword					
1	x_1	6/21	6/21	10/21	1
2	x_2	5/21	5/21	6/21	
00	x_3	4/21	4/21	5/21	
01	x_4	3/21	3/21		
020	x_5	2/21	3/21		
021	x_6	1/21			
022	x_7	0/21			

(c) The expected length of the codewords for the binary Huffman code is $51/21 = 2.43$ bits.

The ternary code has an expected length of $34/21 = 1.62$ ternary symbols.

Question 3.

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:

(a) $\mathbf{p} = (\frac{10}{41}, \frac{9}{41}, \frac{8}{41}, \frac{7}{41}, \frac{7}{41})$

(b) $\mathbf{p} = (\frac{9}{10}, (\frac{9}{10})(\frac{1}{10}), (\frac{9}{10})(\frac{1}{10})^2, (\frac{9}{10})(\frac{1}{10})^3, \ldots)$

Solution: *Data compression*

(a)

Code	Source symbol	Prob.				
10	A	10/41	14/41	17/41	24/41	41/41
00	B	9/41	10/41	14/41	17/41	
01	C	8/41	9/41	10/41		
110	D	7/41	8/41			
111	E	7/41				

(b) This is case of an Huffman code on an infinite alphabet. If we consider an initial subset of the symbols, we can see that the cumulative probability of all symbols $\{x : x > i\}$ is $\sum_{j>i} 0.9 * (0.1)^{j-1} = 0.9(0.1)^{i-1}(1/(1-0.1)) = (0.1)^{i-1}$. Since this is less than $0.9 * (0.1)^{i-1}$, the cumulative sum of all the remaining terms is less than the last term used. Thus Huffman coding will always merge the last two terms. This in terms implies that the Huffman code in this case is of the form 1,01,001,0001, etc.

Question 4.

Shannon code. Consider the following method for generating a code for a random variable X which takes on m values $\{1, 2, \ldots, m\}$ with probabilities $p_1, p_2, \ldots, p_m$. Assume that the probabilities are ordered so that $p_1 \geq p_2 \geq \cdots \geq p_m$. 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 $l_i = \lceil \log \frac{1}{p_i} \rceil$.

(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: *Shannon code.*

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

$$\log \frac{1}{p_i} \leq l_i < \log \frac{1}{p_i} + 1$$

which implies that

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

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

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

Thus F_j, $j > i$ differs from F_i by at least 2^{-l_i}, and will therefore differ from F_i is 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 $l_j \geq l_i$, 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.