ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 4 — Information Theory and Coding

Homework 2 — Sep. 23, 2014

PROBLEM 1. Let $\bar{M} = \sum_{i=1} p_i l_i^{100}$ be the 100th moment (i.e., the expected value of the 100th power) of the code word lengths l_i associated with an encoding of a random variable X with distribution p. Let $\bar{M}_1 = \min \bar{M}$ over all prefix-free codes for X; and let $\bar{M}_2 = \min \bar{M}$ over all uniquely decodable codes for X. What relationship exists between $\bar{M}_1$ and $\bar{M}_2$?

PROBLEM 2. Consider the following method for constructing binary code words for a random variable U which takes values $\{a_1, \dots, a_m\}$ with probabilities $P(a_1), \dots, P(a_m)$. Assume that $P(a_1) \geq P(a_2) \geq \cdots \geq P(a_m)$. Define

$$Q_1 = 0 \quad \text{and} \quad Q_i = \sum_{k=1}^{i-1} P(a_k) \quad \text{for } i = 2, 3, \dots$$

The code word assigned to the letter a_i is formed by finding the binary expansion of $Q_i < 1$ (i.e, $1/2 = .100\dots$, $1/4 = .0100\dots$, $5/8 = .1010\dots$) and letting the codeword be the first l_i bits of this expansion where $l_i = \lceil -\log_2 P(a_i) \rceil$.

(a) Construct binary code words for the probability distribution $\{1/4, 1/4, 1/8, 1/8, 1/16, 1/16, 1/16, 1/16\}$.

(b) Prove that the method described above yields a prefix-free code and the average codeword length $\bar{L}$ satisfies

$$H(X) \leq \bar{L} < H(X) + 1.$$

PROBLEM 3. A random variable takes values on an alphabet of K letters, and each letter has the same probability. These letters are encoded into binary words using the Huffman procedure so as to minimize the average code word length. Let j and x be chosen such that $K = x2^j$, where j is an integer and $1 \leq x < 2$.

(a) Do any code words have lengths not equal to j or $j + 1$? Why?

(b) In terms of j and x, how many code words have length j?

(c) What is the average code word length?

PROBLEM 4. Consider two discrete memoryless sources. Source 1 has an alphabet of 6 symbols with the probabilities, 0.3, 0.2, 0.15, 0.15, 0.1, 0.1. Source 2 has an alphabet of 7 letters with probabilities 0.3, 0.25, 0.15, 0.1, 0.1, 0.05, 0.05. Construct a binary ($D = 2$) Huffman code and a ternary ($D = 3$) Huffman code for each source. Find the average number of code letters per source symbol in each case. Hint: observe that a ternary tree has an odd number of leaves. A fictitious symbol with probability 0 might therefore be needed for the code construction.

Problem 5.

(a) A source has an alphabet of 4 letters, a_1, a_2, a_3, a_4, and we have the condition $P(a_1) > P(a_2) = P(a_3) = P(a_4)$. Find the smallest number q such that $P(a_1) > q$ implies that $n_1 = 1$ where n_1 throughout this problem is the length of the codeword for a_1 in a Huffman code.

(b) Show by example that if $P(a_1) = q$ (your answer in part (a)), then a Huffman code exists with $n_1 > 1$.

(c) Now assume the more general condition, $P(a_1) > P(a_2) \geq P(a_3) \geq P(a_4)$. Does $P(a_1) > q$ still imply that $n_1 = 1$? Why or why not?

(d) Now assume that the source has an arbitrary number K of letters with $P(a_1) > P(a_2) \geq \cdots \geq P(a_K)$. Does $P(a_1) > q$ now imply $n_1 = 1$?

(e) Assume $P(a_1) \geq P(a_2) \geq \cdots \geq P(a_K)$. Find the largest number q' such that $P(a_1) < q'$ implies that $n_1 > 1$.

Problem 6. The following problem concerns a technique known as run length coding. Along with being a useful technique, it should make you look carefully into the sense in which Huffman coding is optimal. A source produces a sequence of independent binary digits with probabilities $P(\mathbf{0}) = 0.9$ and $P(\mathbf{1}) = 0.1$. We shall encode this sequence in two stages, first counting the number of 0's between successive 1's in the source output, and then encoding these counts into binary code words. The first stage of encoding maps source sequences into intermediate digits by the following rule:

Source Sequence	Intermediate Digits (# of zeros)
1	0
01	1
001	2
0001	3
⋮	⋮
00000001	7
00000000	8

Thus the following sequence is encoded as follows:

$$\begin{array}{cccccccccccccccccccc} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0, & & & 2, & & & & & & & & 8, & & & 2, & 0, & & & & & 4 \end{array}$$

The final stage of encoding assigns a code word of length 1 to the intermediate digit 8 and codewords of length 4 to the other intermediate digits.

(a) Justify that the overall code is uniquely decodable.

(b) Find the average number $\bar{N}$ of source digits per intermediate digit.

(c) Find the average number $\bar{M}$ of encoded binary digits per intermediate digit.

(d) Show, by appeal to the law of large numbers, that for a very long source sequence of source digits, the ratio of the number of encoded binary digits to the number of source digits will with high probability be close to $\bar{M}/\bar{N}$. Compare this ratio to the average number number of code letters per source letter for a Huffman code encoding four source digits at a time.