Team homework set 3

Your homework must be handed in **electronically via Moodle before Friday November 25th, 20:00h**. This deadline is strict and late submissions are graded with a 0. At the end of the course, the lowest of your 6 weekly homework grades will be dropped. You are strongly encouraged to work together on the exercises, including the homework. However, after this discussion phase, you have to write down and submit your own individual solution. Numbers alone are never sufficient, always motivate your answers.

Problem 1: Huffman Coding

(a) **(4pt)** Consider the binary source P_X with $P_X(0) = \frac{1}{8}$ and $P_X(1) = \frac{7}{8}$. Design a Huffman code for P_X. Describe, in order, which source symbols you combine, and list the final codebook. What is the average codeword length?

(b) **(4pt)** Design a Huffman code for blocks of $N = 2$ bits drawn from the source P_X. Describe, in order, which source symbols you combine, and list the final codebook. What is the average codeword length?

(c) **(4pt)** Design a Huffman code for blocks of $N = 3$ bits drawn from the source P_X. Describe, in order, which source symbols you combine, and list the final codebook. What is the average codeword length?

(d) **(4pt)** For the three codes you designed ($N = 1, 2, 3$), divide the average codeword length by N, and compare these values to the optimal length, i.e., $H(X)$. What do you observe?

(e) **(1pt)** If you were asked to design a Huffman code for a block of $N = 100$ bits, what problem would you run into?

(f) **(2pt)** Consider the random variable Z with

z	1	2	3	4	5	6
$P_Z(z)$	1/10	3/10	2/10	2/10	1/10	1/10

Design a *ternary* Huffman code for Z (i.e. using an alphabet with three symbols).

Problem 2: Inefficiency when using the wrong code

(a) **(2pt)** Given are two distributions P_X and Q_X for the same set $\mathcal{X} = \{a, b, c, d\}$:

x	a	b	c	d
$P_X(x)$	1/4	1/4	1/4	1/4
$Q_X(x)$	1/2	1/4	1/8	1/8

Design an optimal prefix-free code for $Q_X(x)$.

(b) **(4pt)** What is the expected codeword length of the code you just designed? (Three decimals precision)

(c) **(4pt)** What is the expected codeword length if you use this code to encode the source P_X? (Three decimals precision)

(d) **(4pt)** Show that in general, when using optimal code for any source source Q_X (which has lengths $\ell(x) = \lceil -\log Q_X(x) \rceil$ for all $x \in \mathcal{X}$) to encode another source P_X, this incurs a penalty of $D(P_X \| Q_X)$ in the average description length.

More formally, prove that

$$H(P_X) + D(P_X \| Q_X) \leq \mathbb{E}_{P_X}[\ell(X)] \leq H(P_X) + D(P_X \| Q_X) + 1$$

for all P_X and Q_X.

Problem 3: Shannon code (6pt)

Consider the following method for generating a code for a random variable X which takes on m values $\{1, 2, ..., m\}$. Assume that the probabilities are ordered such that $P_X(1) \geq P_X(2) \geq ... \geq P_X(m)$. Define

$$F_i := \sum_{k=1}^{i-1} P_X(k),$$

the sum of the probabilities of all symbols less than i. Then the Shannon code is defined by assigning the (binary representation of the) number

$F_i \in [0, 1]$ as the codeword for i, where F_i is rounded off to $\lceil \log \frac{1}{P_X(i)} \rceil$ bits.

(a) **(1pt)** Construct the code for the probability distribution $P_X(1) = \frac{1}{2}$, $P_X(2) = \frac{1}{4}$, $P_X(3) = P_X(4) = \frac{1}{8}$

(b) **(1pt)** Construct the code for the probability distribution $P_Y(1) = P_Y(2) = P_Y(3) = \frac{1}{3}$.

(c) **(2pt)** Show that the Shannon code is prefix-free.

(d) **(2pt)** Show that the average length ℓ_S of the Shannon code satisfies

$$H(X) \leq \ell_S(P_X) < H(X) + 1.$$

Problem 4: Calculation of the typical set (8pt)

To clarify the notion of a typical set $A_\varepsilon^{(n)}$ and the smallest set of high probability $B_\delta^{(n)}$, we will calculate these sets for a simple example. Consider a sequence of i.i.d. binary random variables $X_1, X_2, ... X_n$, where the probability that $P_X(1) = 0.6$ and $P_X(0) = 0.4$.

(a) **(1pt)** Calculate $H(X)$.

(b) **(3pt)** With $n = 25$ and $\varepsilon = 0.1$, which sequences fall in the typical set $A_\varepsilon^{(n)}$? What is the probability of the typical set? How many elements are there in the typical set? (This involves computation of a table of probabilities for sequences with k 1's, $0 \leq k \leq 25$, and finding those sequences that are in the typical set.)
Hint: Here is the table: http://goo.gl/sQCPM0

(c) **(2pt)** How many elements are there in the smallest set that has probability 0.9? In other words, what is $|B_\delta^{(n)}|$ for $n = 25$ and $\delta = 0.1$?

(d) **(2pt)** How many elements are there in the intersection $|A_\varepsilon^{(n)} \cap B_\delta^{(n)}|$ of the sets computed in parts (b) and (c)? What is the probability of this intersection?

Problem 5: Three random variables (3pt)

Let A, B, C be random variables such that

$$\begin{aligned} I(A; B) &= 0 \\ I(A; C|B) &= I(A; B|C) \\ H(A|BC) &= 0 \end{aligned}$$

Prove one of the following possible relations between $H(A)$ and $H(C)$:

$$\begin{aligned} H(A) &= H(C), \\ H(A) &\leq H(C), \\ H(A) &\geq H(C), \\ H(A) &< H(C), \\ H(A) &> H(C). \end{aligned}$$

Problem 6: Piece of cake (2pt)

A big cake is repeatedly sliced into two pieces. At every step, the *smaller* part is discarded (or eaten). On the other part the process is continued. At every step, the remaining piece is cut randomly into two pieces with the following proportions:

$$\left(\frac{2}{3}, \frac{1}{3}\right) \text{ with probability } \frac{3}{4},$$

$$\left(\frac{3}{5}, \frac{2}{5}\right) \text{ with probability } \frac{1}{4}.$$

For example, three consecutive cuts might result in a piece of cake of size $\frac{3}{5} \cdot \frac{2}{3} \cdot \frac{2}{3}$. Let T_n be the fraction of the cake left after n cuts. Clearly, this fraction T_n decreases exponentially with n, i.e. $T_n \approx c^n$ for some real constant $0 < c < 1$. Determine c for large values of n.

Hint: Let C_i be a random variable describing the fraction of the cake that is cut (and kept) at the ith cut, i.e. $C_i = \frac{2}{3}$ with probability $\frac{3}{4}$, or $\frac{3}{5}$ otherwise. Then use the weak law of large numbers.