Practice problem set 2

This week's exercises deal with Huffman codes, arithmetic codes and the AEP. You do not have to hand in these exercises, they are for practicing only. Problems marked with a ★ are generally a bit harder. If you have questions about any of the exercises, please post them in the discussion forum on Moodle, and try to help each other. We will also keep an eye on the forum.

Problem 1: Prefix-free arithmetic codes

(a) What are the names of the binary intervals $[\frac{6}{8}, \frac{7}{8})$ and $[\frac{7}{16}, \frac{8}{16})$?

(b) What are the binary intervals with the names 0110 and 011?

(c) Prove that if the name of a binary interval I is the prefix of the name of another binary interval J, it must be that $J \subset I$.

(d) Use (c) to prove that for any source, the resulting arithmetic code AC^{pf} is indeed prefix-free.

Problem 2: Non-prefix-free arithmetic codes

In class, we have seen a procedure to build a prefix-free arithmetic code AC for X by dividing $[0, 1)$ into smaller intervals I_x (for $x \in \mathcal{X}$) according to the probability distribution P_X, and picking $AC(x)$ to be the (name of the) largest binary interval that fits into I_x. In this exercise, we consider a simpler procedure that creates slightly shorter codewords, but is not necessarily prefix-free.

(a) Given X with $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ and $P_X(\mathtt{a}) = P_X(\mathtt{b}) = 1/3$, $P_X(\mathtt{c}) = P_X(\mathtt{d}) = 1/6$. Draw the intervals I_x on $[0, 1)$. Then assign codewords to each x by finding a number in each interval with a binary representation that is as short as possible. Note that there are sometimes multiple possibilities!

(b) Also find the prefix-free arithmetic code AC^{pf} for this source. How do the average codeword lengths compare?

(c) Recall the proof that $\ell_{AC^{pf}}(P_X) \leq H(X) + 2$. Adapt the proof to show that for the non-prefix-free procedure, the average codeword length $\ell_{AC}(P_X)$ is upper bounded by $H(X) + 1$ for any source.

Problem 3: Optimal codeword lengths

(CT, Exercise 5.22) Although the codeword lengths of an optimal variable length code are complicated functions of the source probabilities, it can be said that less probable symbols are encoded into longer codewords. Suppose that the message probabilities are given in decreasing order $p_1 > p_2 \geq \cdots \geq p_m$.

(a) Prove that for any binary Huffman code, if the most probable message symbol has probability $p_1 > 2/5$, then that symbol must be assigned a codeword of length 1.

(b) Prove that for any binary Huffman code, if the most probable message symbol has probability $p_1 < 1/3$, then that symbol must be assigned a codeword of length ≥ 2.

Problem 4: Shannon code

(CT, exercise 5.25) 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 su 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) 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) Construct the code for the probability distribution $P_Y(1) = P_Y(2) = P_Y(3) = \frac{1}{3}$.

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

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

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

define shannon code, construct the code for some example distribution, and then show that the length is optimal. (and prefix-free)

Problem 5: AEP and source coding

A discrete memoryless source emits a sequence of of statistically independent binary digits with probabilities $P_X(1) = 0.005$ and $P_X(0) = 0.995$. The digits are taken 100 at a time and a binary codeword is provided for every sequence of 100 digits containing three or fewer 1's.

(a) **(2pt)** Assuming that all codewords are the same length, find the minimum length required to provide codewords for all sequences with three or fewer 1's.

(b) **(2pt)** Calculate the probability of observing a source sequence for which no codeword has been assigned.

(c) **(3pt)** In the second homework problem set, you were asked to prove Chebyshev's inequality. Use it to bound the probability of observing a source sequence for which no codeword has been assigned. Compare this bound with the actual probability computed in part (b).

Problem 6: Calculation of the typical set

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 7: Sampling from any distribution using random bits

In this exercise, we come up with a strategy to sample from an arbitrary distribution P_X using fair random bits (for example, the outcome of a sequence of fair coin tosses).

(a) Let Z_1 be a random variable with $\mathcal{Z}_1 = \{a, b, c\}$ and $P_{Z_1}(a) = 1/2$, $P_{Z_1}(b) = P_{Z_1}(c) = 1/4$. Come up with a strategy to sample from X using a number of fair coin tosses. How many coin tosses do you expect to do? How does this compare to the entropy of Z?

(b) Consider the binary expansion of some $p_i \in [0, 1)$. Let the *atoms* of this expansion be the set $At_i := \{2^{-k} \mid \text{ the } k^{th} \text{ bit of the binary expansion of } p_i \text{ is } 1.\}$. Find the atoms for the binary expansion of $p_1 = \frac{1}{3}$ and $p_2 = \frac{2}{3}$.

(c) Show that for any probability distribution with probabilities $(p_1, ..., p_n)$, it is possible to construct a binary tree (the *sampling tree* for this distribution) such that if $2^{-k} \in At_i$ for some i, then the tree contains a leaf with label i at depth k. **Hint:** use Kraft's inequality.

(d) Let Z_2 be a random variable with $\mathcal{Z}_2 = \{a, b\}$ and $P_{Z_2}(a) = 1/3$, $P_{Z_2}(b) = 2/3$. Construct the sampling tree for P_{Z_2}. Find a fair coin and use it to sample from this distribution, following the strategy described by the sampling tree.

Let $ET(X)$ denote the expected number of coin tosses when sampling from X using the sampling tree described above. In the rest of this exercise, you will show that this method of sampling from an arbitrary distribution P_X using fair random bits is quite efficient in terms of $ET(X)$.

(e) Given a sampling tree for an arbitrary distribution P_X, define a random variable Y with $\mathcal{Y}$ the set of all leafs of the tree, and $P_Y(y) = 2^{-d(y)}$, where $d(y)$ is the depth of the leaf y in the tree. Prove that $H(Y) = ET(X)$.

(f) Use the result from (e) to prove that $H(X) \leq ET(X)$.

★ Prove that $H(Y|X) < 2$ (**Hint:** see Cover and Thomas, Section 5.12)

(g) Use the result from ★ to prove that $ET(X) < H(X) + 2$.