INFORMATION THEORY
Master of Logic, Master AI, Master CS, University of Amsterdam, 2018
TEACHER: Christian Schaffner, TAs: Yfke Dulek, Esteban Landerreche, Kyah Smaal

Practice problem set 3

This week's exercises deal with source codes and data compression. You do not have to hand in these exercises, they are for practicing only. During the work session, start with solving the exercise you will be moderating. Work out a full solution on paper/computer and get it approved by the teacher. Also think about the following questions: What is the point of the exercise? What kind of problems will students encounter when solving this problem? What kind of questions could you ask the presenter on Friday? 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 Canvas, and try to help each other. We will also keep an eye on the forum.

Problem 1: An optimal code

Let X be a random variable.

(a) Show that if there exists an $n \in \mathbb{N}$ such that for all $x \in \mathcal{X}$, $P_X(x) = \frac{1}{2^n}$, then there exists a source code whose expected length equals the entropy.

(b) ([MacKay], Exercise 5.25:) Show that if for all $x \in \mathcal{X}$, there exists an $n \in \mathbb{N}$ such that $P_X(x) = \frac{1}{2^n}$, then there exists a source code whose expected length equals the entropy.

★ Problem 2: Unique decodability

Construct a binary symbol code (for a finite alphabet $\mathcal{X}$ of your own choice) that is uniquely decodable, but for which there exists an *infinite* binary string that can be decoded in more than one way.

Problem 3: 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 4: Non-prefix-free arithmetic codes

In class, we have seen a procedure to build a prefix-free arithmetic code AC^{pf} 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 $AC^{pf}(x)$ 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 $\mathcal{X} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$ and $P_X(\mathsf{a}) = P_X(\mathsf{b}) = 1/3$, $P_X(\mathsf{c}) = P_X(\mathsf{d}) = 1/6$. 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 5: 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_1?

(b) Consider the binary expansion of some $p_i \in [0, 1)$. Let the *atoms* of this expansion be the set $At_i := \{2^{-k} \mid$

the k^{th} bit of the binary expansion of p_i 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 $ET(X) < H(X) + 2$.

★ Problem 6: 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.