Information and coding theory

Project II

Practical information

Each project should be executed in groups of two students. We expect each group to provide:

- A *brief* report (in PDF format) collecting the answers to the different questions.
- The scripts (in Python or Julia) you have implemented.

The report and the scripts should be submitted as a tar.gz (or zip) file on Montefiore's submission plateform (`http://submit.montefiore.ulg.ac.be`) before *May 2, 23:59 GMT+2*. **You must concatenate your sXXXXXX ids as group, archive and report names.**

Questions

Source coding and reversible data compression

Let us consider a source that can emit symbols from an alphabet including all lower-case letters[1] ['a','b',...,'z'], all numbers ['0','1',...,'9'], and some additional characters, i.e. ['.',',','?',';', ...].

A sequence emitted by this source is provided ("text.csv", and hereafter referred to as "the text sample").

1. Determine a set of additional characters and give the total number of possible symbols given this set. Justify your choice(s).
2. Estimate the marginal probability distribution of all symbols from the text sample. What are your assumption(s) on the source model if you only consider the marginal probability distribution?
3. Implement a function that returns a binary Huffman code for a given probability distribution. Explain how to extend your function to generate a Huffman code of any alphabet size.
4. Using your function that implements the Huffman algorithm find an optimal code for the marginal distribution of source symbols. Using this code, encode the original text and give the total length of the coded text sample.
5. Give the expected average length for this code. Is it different from the empirical average length estimated from the coded text sample? Justify.
6. Compute the compression rate between the original text sample and its coded version. Details your procedure and justify your result theoretically.
7. How could you improve the source model used so far? Does this necessarily improve the compression rate? Implement this solution and give the new compression rate. Justify your answers.

[1] Note that upper-case letters should be transformed into lower-case ones.

Channel coding and irreversible data compression

Let us now consider a sound signal that is sent through a noisy channel. This sound signal is first coded in a binary alphabet and then each binary symbol is sent through a binary symmetric channel with a probability of error equal to 0.01.

Let us consider the .wav file "sound.wav" as the sound signal. Its sampling resolution is such that possible values are between 0 and 255, and its sampling rate is $11025Hz$.

8. Give the plot of the sound signal and listen to it.
9. Transform the sound signal using a (naive) fixed-length binary code. What is the appropriate codeword length? Justify.
10. Implement a function that returns the Hamming (7,4) code for a given sequence of binary symbols. Using your function, code the binary sound signal (obtained in the previous question).
11. Simulate the channel effect on a sequence of symbols and generate a corrupted version of the coded binary sound signal. Plot and listen to the corrupted signal. What do you notice?
12. Using the property of the Hamming code, try to recover the corrupted binary sound signal. Explain your procedure.
13. How would you proceed to reduce the loss and/or to improve the compression rate? Justify.

Image compression

The given script "image_compression.py" applies the cosine transform on any square image[2] in order to get a set of coefficient values. An image is then reconstructed with a subset of these coefficients.

14. What are image transformations used for in the context of data compression? Justify and give at least one example.
15. What are image transformations also used for (except for data compression)? Justify and give at least one example.
16. What is the interest of keeping only (the values of) some coefficients to reconstruct the image?
17. Which coefficients should be kept? And how many of them? Discuss.
18. How could you increase the compression rate even more? Discuss.

[2] You can use either the given "image.png" or any other square .png image.