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 the deadline. **You must concatenate your sXXXXXX ids** (e.g., s000007s123456) **as group, archive and report names.**

Questions

Source coding and reversible data compression

Let us consider a source S using a Q-ary alphabet. A sequence of source messages (i.e., emitted by this source) is provided ("text.csv", and hereafter referred to as "the text sample").

1. Determine the set of source symbols of S from the text sample. In particular, determine the value of Q. Justify your choice(s).
2. Estimate the marginal probability distribution of all Q 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. Give the main steps of your implementation. 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. Give the total length of the coded text sample and its (empirical) average length.
5. Give the expected average length for this code. Compare this value with (a) the empirical average length, and (b) theoretical bound(s)? Justify.
6. Compute the compression rate between the original text sample and its coded version. Detail your procedure and discuss your result.
7. Let us assume that you use the relative frequency of letters (and symbols) in the English language instead of a marginal probability distribution estimated from the text sample. What does it change? What if we consider relative frequencies in another language?
8. Let us assume that you want to reduce the size of alphabet Q. Imagine ways of reducing Q without losing too much (textual) information. Discuss.
9. How could you improve the source model used so far? Give several ideas and explain how they actually improve the source model. Justify your answers.

10. How could you increase even more the compression rate of the Huffman code with respect to the original source message? Give at least one idea and justify.

11. Let us now consider the text sample in a binary alphabet ("binary_text.csv" is made of binary representations (8-bits) of extended-ASCII symbols). Implement a function that encodes this binary text sample using an on-line Lempel-Ziv code (as seen in the theoretical course).

Reversible image compression

Let us consider an uncompressed greyscale image (i.e., one value per pixel). The PNG format encoding combines a dictionary method (LZ77) and a Huffman code.

12. Explain how the dictionary method and Huffman code can be combined.

13. Implement a function that converts an uncompressed greyscale image (i.e., one value per pixel) in the PNG format. Apply this function on the given uncompressed image ("lena512.mat") and give the compression rate.
Suggestion: You can use (or adapt) the function implemented at question 11 and you can omit headers that are required in the actual PNG file format.

14. Discuss the settings where (a) the dictionary method, (b) the Huffman code and (c) the PNG compression technique should be the most efficient.

Channel coding

Let us consider a sound signal that is sent through a noisy channel. Let us take a .wav file "sound.wav" as sound signal. Its quantisation is such that possible values are between 0 and 255, and its sampling rate is $11025Hz$. The channel is a binary symmetric channel with a probability of error equal to 0.01.

In order to send the sound signal through the channel, the signal is first encoded in a binary alphabet and then each binary symbol is sent through the channel.

15. Give the plot of the sound signal and listen to it.

16. Encode the sound signal using a fixed-length binary code. What is the appropriate number of bits? Justify.

17. Simulate the channel effect on the binary sound signal. Then decode the sound signal. Plot and listen to the decoded sound signal. What do you notice?

18. Instead of sending directly through the channel the binary sound signal, you will first introduce some redundancy. To do that, implement a function that returns the Hamming (7,4) code for a given sequence of binary symbols. Then, using your function, encode the binary sound signal (from question 16).

19. Simulate the channel effect on the binary sound signal with redundancy. Then decode the binary sound signal. Plot and listen to the decoded sound signal. What do you notice? Explain your decoding procedure.

20. How would you proceed to reduce the loss of information and/or to improve the communication rate? Justify.