Information and coding theory

Project (part III)

Deadline: 27 April 2018

Follow the policy for plagiarism[1] for every part of this project or you expose yourself to measures taken by the ULiège against plagiarism[2]. By submitting a report, you agree with those terms.

Description of the problem

This third part of the project aims to help you to become familiar with source coding. Some theoretical notions will be illustrated and applied on a medical database and a short text.

A bit of theory

This part mainly requires to apply the notions seen during the theoretical course about source coding. In particular, it is advised to revise the problem definition of a discrete source coding and the Huffman's code.

Practical informations

Realisation

Let define two sources:

- $\mathcal{S}_1$ is a memoryless stationary source. The database *data* is a message transmitted by this source. Each symbol transmitted by the source is made of values of four variables and corresponds to results of a patient for four tests. Let's assume that the alphabet used for each symbol is the ten Arabic numerals $(0, 1, \ldots, 9)$.

- $\mathcal{S}_2$ is another source. A sample of message *text_ sample* transmitted by this source is provided. For the sake of simplicity, all uppercase letters have been replaced by the corresponding lowercase letters.

In a more practical way, besides the database *data*, and the short text *text_ sample*, you have the probability distribution of each symbol transmitted by the source $\mathcal{S}_1$. You may also find interesting some functions converting numbers to strings (e.g., **int2str** or **num2str**) or strings to numbers (e.g., **str2num**). Indeed, one simple way to carry out some steps of this work is to manipulate strings (i.e., chains of characters).

Submission

You have to answer all questions in a report (**of maximum 5 pages**) and submit everything (report and codes) before the deadline using the Montefiore Submission plateform (`http://submit.montefiore.ulg.ac.be/`). This project is individual and all pieces of code and all answers should be personnal.

[1] `https://www.ulg.ac.be/cms/c_146131/en/what-are-the-various-forms-of-plagiarism`

[2] `https://www.ulg.ac.be/cms/c_479794/en/the-measures-taken-by-the-ulg-against-plagiarism`

Questions

Implementation

1. Write a function **result = all_symbols(cardinality)** which takes a vector of size p containing the cardinalities of each one of p variables and returns a line vector of size N with all N possible symbols (as strings) according to the encoding method used for the given database *data*, ordered in ascending order.
   Example: ***all_ symbols([2,3])*** *returns [11 12 13 21 22 23].*

2. Write a function **result = estimate_proba(message, alphabet)** which takes a message transmitted by the source $\mathcal{S}_1$ (i.e., a line vector of length N with a separator between two consecutive symbols) using an alphabet of n symbols listed in *alphabet* and returns a line vector of size n such that *result(i)* is the estimated probability of symbol *alphabet(i)*.

3. Write a function **result = encode_to_numerals(data)** which encodes the database by associating each symbol to a natural number. The natural numbers are associated with symbols in ascending order starting from 1.
   Example: ***encode_ to_ numerals([13 12 11 21 23 22])*** *returns [3 2 1 4 6 5].*

4. Write a function **result = Huffcode(probabilities)** with takes a vector of length N containing the probability distribution of the different symbols and returns a vector containing encoded symbols according to Huffman's code, in a binary alphabet. Give the main steps of your implementation.
   Example: ***Huffcode([0.25 0.5 0.25])*** *returns [10 0 11] (or equivalently [21 1 22]).*

Application

Let's talk about text.

5. Let's consider $\mathcal{S}_2$ as a memoryless stationary source and the given text as a message transmitted by this source. Would you choose the English alphabet as alphabet for this source? Do you think another alphabet can be more appropriate? Give the probability distribution of each symbol and the entropy of the source. What can you say about the information brought by a symbol? Justify (theoretically).

6. Let's still assume that the source $\mathcal{S}_2$ giving the text is memoryless and stationary. Do you think that other possibilities exist to model this database? If such possibilities exist, give an example for each one. Discuss and justify in every case.

7. Let's now assume that we have no assumptions about the source $\mathcal{S}_2$ and you only have the transmitted message. How would you model the source? What are the motivations and the assumptions behind your choice? Does this new model change something in terms of coding theory? Justify and discuss.

Let's talk about data.

8. What is the length of each symbol transmitted by the source $\mathcal{S}_1$? Is Kraft's inequality verified by this code and what can you say about that? Is this code complete? What is the optimal average length for one symbol? Justify.

9. Calculate the compression rate and the length expectation of a symbol obtained with the database *data* encoded by the function **encode_to_numerals** and denoted *encoding1*.

10. Is Kraft's inequality verified by this code? Is it decodable, complete, regular, instantaneous, prefix-free, absolutely optimal? Justify.

11. Give Huffman's encoding corresponding to the given probability distribution. Calculate the compression rate for a binary Huffman's coding applied on the database model in relation to both previous encodings. What is the length expectation of a symbol transmitted by the source using the binary Huffman's coding? Justify.

12. What is the optimal average length of a symbol? Is the code decodable, complete, regular, instantaneous, prefix-free, absolutely optimal? Justify.

13. Encode the given database with Huffman's code (*encoding2*). What is the obtained average length? What do you notice in comparison with question 11? Justify.

14. What alphabet sizes can be used to create a complete code for this source? Justify.