Hand in problem 2 in Information Theory (EITN45)

Spring 2020

Problem

In this problem a text should be compressed using Huffman coding. The text is a well used reference text in source coding, *Alice's adventures in wonderland* by Lewis Carroll. The version used here is from `http://corpus.canterbury.ac.nz/` as `Alice29.txt`, in the Canterbury Corpus. It is given with the same name at the course instant at Canvas. To solve the problem you first need to estimate the probability distribution for the letters included in the file, and then construct an optimal source code based on this estimation. Use the code to compress the text. In your answer you should compare the average codeword length with the entropy of the estimated distribution and with the uncompressed case.

Source

The first step in the problem is to estimate the source distribution of the letters. The size of the text file is 152 089 characters[1] and, apart from normal letters and numbers, it contains characters like space, line feed (LF), carriage return (CR), !, (,), ? etc. In the ASCII table for text files there are in total 256 characters, and in the text 74 different characters are used. (As a check of your derivations, there are 13 381 occurrences of the character `e`. Notice, this is only small `e` and not `E`).

Optimal code

Construct an optimal binary source code for the estimated probability distribution above. Derive the total length of the encoded text in the file Alice29.txt, and compare with the uncoded case (ASCII representation). Also, compare the average codeword length per character with the entropy of the distribution.

[1] Depending on how *new line* is represented it can be either 152 089 or 148 481 characters. Typically Windows uses two characters while UNIX-like systems use one character to represent new line. But it can also differ between programming environment. For example, for the installations I have on my MAC, Python2 uses two characters while Python3 uses one character. Treat each character individually when deriving the Huffman code.

Hand in details

You should hand in your solution to the problem as a pdf in Canvas. This can be written on computer or a scanned copy of hand written papers. In the handed in solution the line of reasoning should be clear from the solution and it should contain

- Distribution for the letters
- Code table and comparisons above
- If you use computer tools to solve parts of the problem you should also hand in the code for these scripts as appendix. (As text in the pdf, we do not want to run any code).

It is important that the solution explains in a clear way how you have solved the problem and what the results are.

Do not forget to write your name and student ID on the handed in papers.

Programming

Since it is a relatively large file it is highly recommended to use computer aid, at least when estimating the distribution. You can use any programming language e.g. Matlab, Octave, Python, R, C/C++, Java, Scala. As long as your program is relatively well structured, we believe we can read most normal languages.