Data Compression
EECS 769: Information Theory
Fall 2018

In this project you will apply source coding to compress an ASCII (text) file. The ASCII file contains a series of characters. In the file, these are stored as 8-bit values; however, the characters are all drawn from the first 128 ASCII values, so they could also be viewed as 7-bit values. The ASCII file is available at:
    http://people.eecs.ku.edu/~perrins/class/F18_769/text.txt
The project has the follwing parts:

1. *Huffman Code.* Apply the Huffman code technique, which is discussed in Chapter 5 of the text. Note that this code requires knowledge of the probability mass function (pmf) of the source alphabet. Use a histogram of the ASCII characters to estimate the pmf.
2. *Shannon–Fano–Elias Code.* Apply the Shannon–Fano–Elias code technique, which is also discussed in Chapter 5 of the text. Here again, this code requires knowledge of the pmf of the source alphabet. Use the same histogram technique as with the Huffman code.
3. *Adaptive Huffman Code.* Apply the adaptive Huffman code technique, which is described in Gallager's 1978 paper entitled "Variations on a theme by Huffman." Note that this code does *not* require advance knowledge of the source pmf, but instead estimates this during algorithm execution.
4. *Tree-based Lempel–Ziv Code.* Apply the tree-based Lempel–Ziv code technique, which is discussed on pages 440–443 of the text.

Write a report summarizing the results of your project.

- For each code, give the first dozen-or-so entries of your codebook. Explain why these entries are the entries you used.
- For each code, give the compression ratio achieved with the code.
- Compare the compression ratios achieved with each of the codes. Explain why your results are correct.
- Compare and contrast the coding algorithms and their performance.