Experiment 3 Variable length source coding

1. Purpose:

1.1 Understanding the distortionless source coding theorem.
1.2 Understanding the Shannon code, Fano code and Huffman code.
1.3 Understanding the metric for source coding.

2. Principle:

The following formulas may be useful during the experiment.

2.1 Average coding length

Single symbol source: $\bar{L} = \sum_{i=1}^{q} P(a_i) l_i$

N times extended source: $\frac{\bar{L}_N}{N} = \frac{1}{N} \sum_{i=1}^{q^N} P(\alpha_i) l_i$

2.2 Coding efficiency

Single symbol source: $\eta = \frac{\frac{H(S)}{\bar{L}}}{\log r}$

N times extended source: $\eta = \frac{\frac{H(S)}{\frac{\bar{L}_N}{N}}}{\log r}$

2.3 Distortionless source coding theorem

$$\frac{H(S)}{\log r} \le \frac{\bar{L}_N}{N} \le \frac{H(S)}{\log r} + \frac{1}{N}$$

3. Procedure:

3.1 Preprocess of the English data

We utilize the same English text data set as in Experiment 1. Differently, we focus on source coding in this experiment. For simplicity, we provide the data after preprocessing in the file "orderlist.mat", note that the data is a 1×39784 vector, where 39784 denotes the number of English letters in the data set, and the value of each element in the vector represents different English letters.

3.2 Source coding for single symbol source

1) Before coding, we first evaluate the probability space of the English letters for a single symbol source, i.e., the 1×27 probability vector. One can use the functions in Experiment 1 for this step. After that, we sort the probabilities in descending order by using the matlab function "sort()" with the argument "descend".
2) We now consider constant length code. What is the (average) coding length for the constant length code with 27 different symbols? If we know the entropy rate of English data is *H(S) = 1.4* bit/sym, what is the coding efficiency?

3) Use the matlab function “SBE()” to code the English data by Shannon code, where the input is the probability vector in descending order, and the output is the strings of codes corresponding to the input probability vector. Evaluate the average coding length and coding efficiency.
   Hints: One can use the matblab function “strlength()” to evaluate the length of the output codewords.
4) Use the matlab function “fano()” to code the English data by Fano code, where the input is the probability vector in descending order, and the output is the strings of codes. Evaluate the average coding length and coding efficiency.
5) Use the matlab function “huffmandict()” to code the English data by Huffman code, where the input is the symbols and the probability vector in descending order. The output is the strings of codes and the average coding length. Evaluate the coding efficiency.
   Hints: For simplicity, we can use numbers 1-27 to represent the input symbols.

3.3 Source coding for 2 times, 3 times and 4 times extended sources

We follow the same steps in 3.2 for different extended sources.

1) Recall that in Experiment 1, the probability space of the English letters for an N times extended source can be evaluated by a $27 \times 27 \cdots \times 27$ matrix. Here, we need to further reshape the probability space matrix into a probability vector, such that it can be used as inputs for the source coding functions. Using matlab function “reshape(p,1,[])” for the reshaping operation.
2) After reshaping, we need to delete the symbols with 0 probability, i.e., the source will not generate those symbols and hence does not need to encode. The elements with zero probability in the probability vector can be deleted by “p(find(p==0))=[]”.
3) Evaluate the average coding length and coding efficiency for constant length code, Shanno code, Fano code and Huffman code respectively.

4. Report Requirements:

4.1 Plot the average coding length against the number of extended times for different coding methods.

4.2 Plot the coding efficiency against the number of extended times for the different coding methods.

4.3 Discuss the observations and conclusions drawn from the plotted figures according to the distortionless coding theorem.