Digital Communication
Assignment II
Source Coding for Markov sequences
Lempel-Ziv and Huffman

Manideep Mamindlapally
(17EC10028)

Aim

Our objectives through this study follow as
(1) Construct a Probability Transition matrix for a discrete Markov process.
(2) Generate a corresponding random Markov sequence.
(3) Perform a Lempel Ziv encoding of the sequence.
(4) Perform Steady state Huffman and Morkov Huffamann encoding.
(4) Compare the results.

Outline of the procedure

- For the purpose of our study an eight element symbol space of `'abcdefgh'` was chosen. $M = 8$
- A Probability transition matrix T was generated by taking a random MxM matrix and dividing each row by its sum in `markov_PTM_generate.m`. The matrix thus generated was saved as `TRANSITION_PROB.mat`.
- A random sequence of length $n = 1000000$ is generated for the above PTM using the Monte Carlo method bin `markov_sequence_generate.m` and saved as `data.txt`.
- The sequence is encoded using the standard Lempel Ziv algorithm in `lempel_ziv_encode.m` and saved as `lempel_ziv_coded_seq.txt`.
- The sequence is encoded using the standard Huffman encoding scheme for the steady state probability distribution π in `huffman_steady_state.m` and saved as `huffman_steady_coded_seq.txt`.
- The sequence is encoded using the modified Huffman encoding scheme for markov sources in `huffman_markov.m` and saved as `huffman_markov_coded_seq.txt`.
- Certain important quantities such as the entropy of steady state probability distribution $H_\pi(X)$ and the markov steady state entropy $H_\infty(X)$ were determined from the `markov_h_pi.m` and `markov_h_inf.m` functions respectively.
- The compression ratio $\mathcal{X}$ for each of the encoding schemes is determined by dividing the length of the binary encoded sequence with the total sequence length. Let us call this quantity $\mathcal{X}_{lempel}$, $\mathcal{X}_{H_{steady}}$ and $\mathcal{X}_{H_{markov}}$ respectively for the three encoding schemes described above.

Results

The following sequence of commands were run at the end of executing `main.m`

```
>> markov_h_pi(T)
       2.0544
>> markov_h_inf(T)
       1.8561
>> lempel_code_length / seq_length
       1.1712
>> huff_steady_code_length / seq_length
       3.0000
>> huff_markov_code_length / seq_length
       2.7500
```

The results could be summarised as $H_\pi(X) = 2.0544$, $H_\infty(X) = 1.8561$, $\mathcal{X}_{lempel} = 1.1712$, $\mathcal{X}_{H_{steady}} = 3.0000$ and $\mathcal{X}_{H_{markov}} = 2.7500$.

Discussion

- It is to be noted that all the three encoding schemes used here are symbol wise binary mappings. A collective symbol mapping would be equivalent to increasing the symbols space length to the corresponding power.
- The entropy of a particular probability distribution would give an estimate of the average number of bits required to encode each source symbol. In the experiment this measure is calculated for two different probability distributions. $\mathcal{X}_{H_{steady}}$ would correspond to the steady state probability distribution while $\mathcal{X}_{H_{markov}}$ would correspond to steady state sampling of symbol wise probability distribution from the PTM.
- According to Shannon, the entropy value is the best achievable theoretical measure of this compression ratio. We have verified the validity of the statement for the Huffman steady state and the Huffman Markov encoding schemes.

$$H_\pi(X) \leq \mathcal{X}_{H_{steady}} < H_\pi(X) + 1$$
$$H_\infty(X) \leq \mathcal{X}_{H_{markov}} < H_\infty(X) + 1$$

- The Shannon's statement however ceases to be true for the Lempel Ziv coding scheme. The compression ratio observed $\mathcal{X}_{lempel}$ was much lower than either of the entropy measures.
- The reason for the above is due to the fact that Lempel Ziv coding is a non entropy based coding unlike the other two. An entropy coding would utilise the symbol wise characteristics and their dependencies at most to the immediate preceding symbol. Lempel Ziv scheme on the other hand utilises the sequence characterstics of the symbol occurences. This would mean a dependency study at a greater length. For this purpose though, it has additional requirements like a huge buffer memory.
- For the same reasons Lempel Ziv code is not guaranteed to work better than the entropy measure for all possible random sequences. For a sequence with no dependencies, for instance a uniform distribution, Huffman coding techniques would work better. The Lempel Ziv performance improves with an increase in skewness of the distribution.