LEMPEL ZIV 77 IMPLEMENTATION ON KNOWN MARKOV SOURCE
ELL712

Submitted by:

Dibyajyoti Jena
2022EEE2712

Submitted to:

Prof. Abhishek Dixit

Department of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY, Delhi

Objectives of the project:

1. Creating a dependent source model with a known entropy (defined Markov Source with known symbol probabilities)
2. Implementation (encoding and decoding) of the Lempel Ziv algorithm.
3. Proving that the Lempel Ziv algorithm achieves an L_{avg-min} close to the lower bound H(X/S) (conditional entropy)

Characteristics of LZ77 Compression Algorithm

1. It is a universal data compressor which operates without prior knowledge of source statistics.

2. Uses variable to variable length codes, in which number of string symbols and number of encoded bits per symbol are variable.

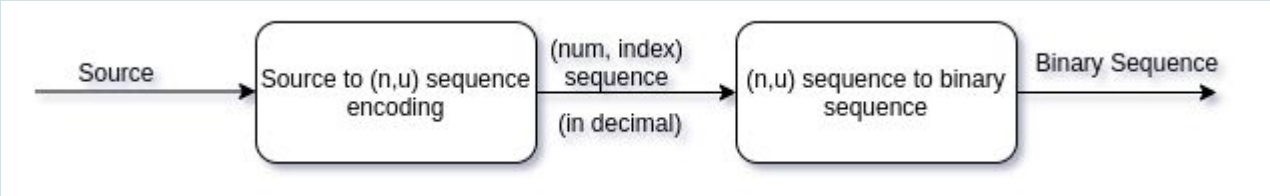
3. Attempts to model the source and encode it simultaneously, with instantaneous decoding.

Setup for the experiment

- LZ77 is to be implemented as taught in the ELL712 course, except a few minor changes for programming convenience. This has been discussed in length in the code report. One such example is window sliding, which helps the algorithm to adapt to with changing source statistics. Since our source is a user defined markov chain with constant statistics, the window was kept constant.

Setup for the experiment

Let x[1], x[2]...x[n] be output string of our defined Markov source. M is the alphabet length(number of distinct symbols present). A window is taken from the string, with length as a power of 2. Symbols in the window are encoded without any compression, since length of the window is considered negligible compared to the string length. The rest of the string is considered as a source sequence and encoded/decoded along this flow-chart:



Implementation Steps:

Implementation of the whole project is in 3 parts:

1. Markov Chain Generation
2. Encoding
3. Decoding

To check:

- **Initial string matches output string**
- **L_{min-avg} approaches conditional entropy of the markov chain H(X/S).**

Encoding and Decoding has been followed as per the course materials and are extensively discussed in the attached code report. Here we will focus more on Markov Chains and the simulation results.

Generation of symbols from a Markov Source

First we need to defined state probabilities **B** and transition probabilities **A**.

Let there be 3 symbols in the sequence generated. Assume each output character is a state, ie, ‘a’, ‘b’ and ‘c’.

Then state probability can be defined as probability of generation of each state irrespective of previous state, given by **B** = [‘a’ ’b’ ’c’] = [0.3 0.4 0.3].

Transition probability, as the name suggests gives probability of occurrence of a symbol given a previous state, given by a matrix: **A** =

	a	b	c
a	0.6	0.3	0.1
b	0	0.7	0.3
c	0.2	0.3	0.5

Generation of symbols from a Markov Source

A =

	a	b	c
a	0.6	0.3	0.1
b	0	0.7	0.3
c	0.2	0.3	0.5

From this matrix, we understand that probability of occurrence of ‘a’ given a state of ‘a’,’b’,’c’ are 0.6, 0 and 0.2 respectively.

After a large sequence is generated, the state probabilities slowly approach stationary values called steady state probabilities or stationary probabilities. In our case, steady state probability vector is given by **B’ = [** 0.1667 0.5 0.3333**],** verified in code report.

Transition probability matrix of Markov source with 5 symbols:

A =

	a	b	c	d	e
a	0.4	0.2	0.3	0.1	0
b	0	0.6	0.1	0.2	0.1
c	0.2	0.2	0.5	0.1	0
d	0.2	0.2	0	0.5	0.1
e	0	0	0.3	0.1	0.6

State Probability Vector **B** = [0.2 0.2 0.3 0.2 0.1]

Steady State Probability Vector **B'** = [0.1455 0.2911 0.2215 0.2151 0.1265]

Entropy

After a string sequence is generated, average of log pmfs of the symbols gives the Entropy H(X), where X is the random variable denoting symbol. H(X) shows uncertainty of a symbol.

Conditional Entropy

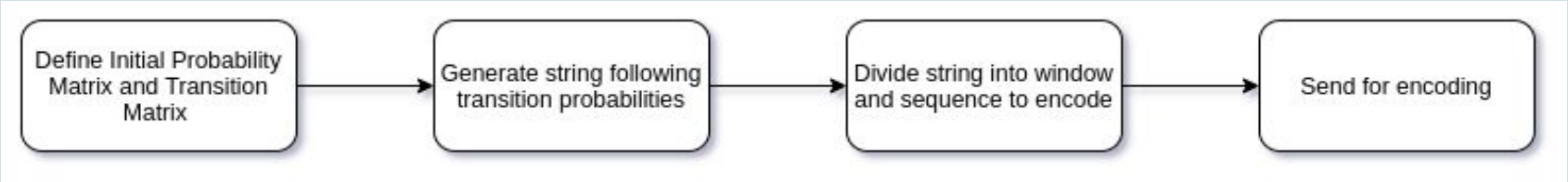
Denoted by H(X/S) where X is the symbol and S is the previous state, showing uncertainty of a symbol given a previous state is known. **Thus theoretically H(X/S) is less than or equal to H(X).**

H(X/S) = $\sum_s \sum_x q(s).p(x/s) log_2 \frac{1}{p(x/s)}$

Where q(s) is steady state probability

p(x/s) denotes transition probability from s to x.

Flowchart For Markov Sequence Generation:

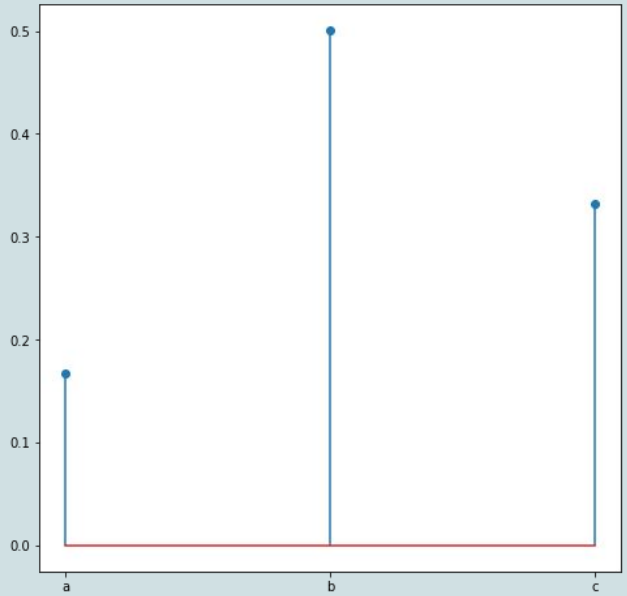


Comparing Results between Markov Sequences with 3 and 5 symbols :

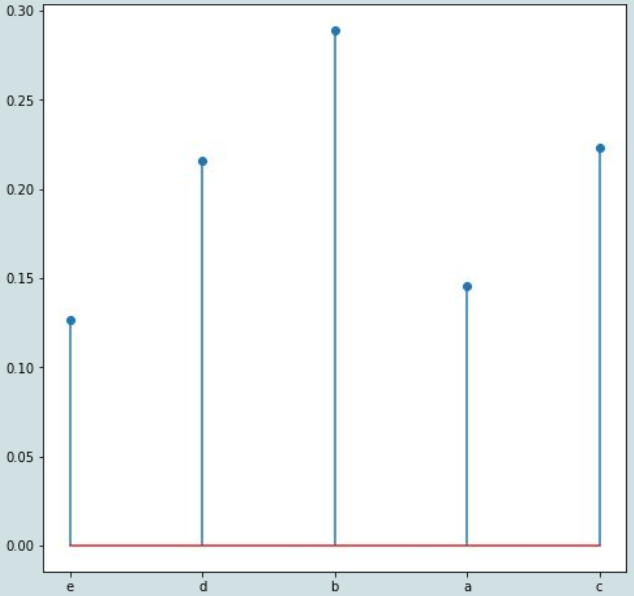
Symbols	Entropy	Conditional Entropy
a,b,c,d,e (1 million length)	2.2588	1.6591
a,b,c (1 million length)	1.4584	1.1517

Taken from code report:
Hence in both cases the Conditional Entropy is less than Entropy measured on the sequence.

Comparing Results between Markov Sequences with 3 and 5 symbols:



Pmf of a 3 symbol Markov sequence



Pmf of a 5 symbol Markov sequence

Both conform to the fact that probabilities tend towards stationary probabilities with increasing length.

Comparing Results between Markov Sequences with 3 and 5 symbols:

Symbols	Total String Length	Window Length	L_avg	Conditional Entropy
3	400,000	2^18	1.88	1.1517
3	1000000	2^18	1.747	1.15
5	400,000	2^18	2.77	1.66
5	1000000	2^18	2.5201	1.66

The above results indicate that compression is happening but L_avg is not approaching the conditional entropy value that it should theoretically achieve in the above configurations.

References:

1. Robert Gallager, course materials for 6.450 Principles of Digital Communications I, Fall 2006. MIT OpenCourseWare(http://ocw.mit.edu/), Massachusetts Institute of Technology.
2. https://numpy.org/doc/stable/