ENEE5304, INFORMATION AND CODING THEORY

Course Project on Source Coding

(You may choose any one of the projects proposed in this set of slides)

Due: December 15, 2021 (via ITC)

Course Project

- You may select any of the projects suggested and explained in class.

The Written Report

- 3-4 pages, double space, 12-point font.
- At least two recent references.
- Write the report in your own words. Do not just copy and paste. If you quote something, cite the reference
- Sections: Define the problem in the introduction, Method (or theoretical background), Results (or Simulations or implementation) and their analysis, the code (appendix), Conclusions, and References.

Presentation

- Students will be required to present their work in my office at designated dates, to be announced later.

Background: Huffman Code (Example)

- Encode the following short text using Huffman encoding

Eerie eyes seen near lake.

- *The sentence has 26 characters. Their frequency of occurrence is*

Char	Freq.	Char	Freq.	Char	Freq.
E	1	y	1	k	1
e	8	s	2	.	1
r	2	n	2		
i	1	a	2		
space	4	l	1		

- *The probability of occurrence of each character can be determined and will be used in the Huffman code.*
- *P(E) = 1/26, P(e) = 8/26, P(space) = 4/26, P(.)=1/26*

Background: Huffman Code (Example)

Symbol	Frequency	Codeword
e	8	11
	4	011
n	2	000
r	2	001
s	2	010
a	2	1001
.	1	10000
E	1	10001
i	1	10100
k	1	10101
l	1	10110
y	1	10111

Summary of Results

H=3.16

$\bar{L}$ **=3.23** bits/character.

Total number of bits in message =**84 bits**.

If ASCII code is used, we need 26*8= **208**

Compression:

$$\frac{\mathbf{84}}{\mathbf{208}} * \mathbf{100\% = 40.36\%}$$

Sentence: Eerie eyes seen near lake.

Code: **1000111001**.....**1010110000**

Course Project 1 on Huffman Code

- Write a computer program using matlab (or any language) to simulate the Huffman code, i.e., to generate the codewords given a certain set of symbols along with their probabilities.
- You will be given an English short story: **Shooting an Elephant by George Orwell.** Find the frequency of the characters in the story.
- Find the probabilities of the characters in the story (do not distinguish between capital and small letters)
- Use your program to find the codewords for the characters.

1. Find the average number of bits/character for the whole story
2. Find the entropy of the alphabet.
3. If ASCII code is used, find the number of bits needed to encode the story.
4. Find the percentage of compression accomplished by using the Huffman encoding as compared to ASCII code.

Course Project 1 on Huffman Code

- What are the probabilities, the lengths of the codewords, and the codewords for the following symbols

Symbol	Probability	codeword	Length of codeword
a			
b			
c			
d			
e			
f			
g			
h			
space			
. (dot)			

Background: LZ Encoding of Binary Data (Example)

***Example** (from Proakis):*

Let us assume that we want to parse and encode the following sequence:

0100001100001010000010100000110000010100001001001 **49 Binary digits**

Parsing the sequence by the rules explained before results in the following phrases:

0, 1, 00, 001, 10, 000, 101, 0000, 01, 010, 00001, 100, 0001, 0100, 0010, 01001, ... **Parsing, 16 phrases**

It is seen that all the phrases are different and each phrase is a previous phrase concatenated with a new source output. The number of phrases is 16. This means that for each phrase we need 4 bits, plus an extra bit to represent the new source output. The above sequence is encoded by

16 phrases require 4 bits (head) + Extra bit for new source symbol (tail) ⇒ 5 bits/phrase

Background: LZ Encoding of Binary Data (Example)

0000 0, 0000 1, 0001 0, 0011 1, 0010 0, 0011 0, 0101 1, 0110 0,
0001 1, 1001 0, 1000 1, 0101 0, 0110 1, 1010 0, 0100 0, 1110 1, ...

Total number of bits in encoded message =16 phrases * 5 bits/phrase = 80 bits

Original message = 49 bits

Dictionary Location		Dictionary Contents	Codeword	
1	0001	0	0000	0
2	0010	1	0000	1
3	0011	00	0001	0
4	0100	001	0011	1
5	0101	10	0010	0
6	0110	000	0011	0
7	0111	101	0101	1
8	1000	0000	0110	0
9	1001	01	0001	1
10	1010	010	1001	0
11	1011	00001	1000	1
12	1100	100	0101	0
13	1101	0001	0110	1
14	1110	0010	1010	0
15	1111	0010	0100	0
16			1110	1

Error here ←

*** Note: 49 data bits are encoded into 80 bits**

*** Question: Where does the compression come from?**

Answer: In short sentences, a saving can hardly be noticed. But in a long text, many phrases of longer lengths become more frequent, and as such these long phrases will be encoded into smaller number of bits.

0100
0010
01001

Course Project 2 on Lempel-Ziv Encoding of Binary Data

1. Generate a random sequence of binary data such that P(1)=0.95 and P(0)=0.05.
2. First, let the size of the sequence be N=100 digits.
3. Develop a program to parse the data and assign a number to each phrase.
4. Find the different phrases of the encoded sequence and the binary digits needed to represent each phrase (the head + tail). **Submit the result in your report**
5. Find the number of bits $\mathbf{N_B}$ needed to represent the 100 bits.
6. Find the compression ratio ($\mathbf{N_B}$ /100)

Course Project 2 on Lempel-Ziv Encoding of Binary Data

1. Repeat the above calculations for a random sequence of sizes as given in the table below.
2. Compare limit on the compression ratio $\mathbf{N_B}$ /N to the source entropy.

Sequence length N	Size of encoded sequence (N_B)	Compression ratio N_B /N	Number of bits per codeword
100			
500			
1000			
1500			
2000			
2500			
3000			
5000			
10000			
20000			

Project 3: Huffman Encoding of the Markov Source

- Consider the Markov source with state diagram as shown in the figure.
- If a message of size 1 symbol is taken, find the Huffman code and the average number of bits per codeword
- If a message of size 2 symbols is taken, find the Huffman code and the average number of bits per codeword

- If a message of size 3 symbols is taken, find the Huffman code and the average number of bits per codeword
- Compare the above results to the the source entropy