

MAA 507 - Mathematics of Internet

Seminar 1 Written Assignment

1) Elif Cemre Durgut
Group 3

2) Discussion questions:

Part1

1) The number we picked from the interval is 0.402, and the decimal part of it is 402 which is in between 256 (2^8) and 512 (2^9). 9 digits would be enough to encode it as a binary number.

2) Currently, we have 5 characters (a, b, c, d, _), so the size of the probability table is 5. If we had conditional probabilities instead of exact probabilities:

1) prob of a following b	}	$C(5, 2) * 2 = 20$
2) prob of b following a		
3) prob of a following c		
4) prob of c following a		
.....		

3) One of the real-world applications of arithmetic encoding is text/image compression.

Part2

1) We managed to decode Elias Gamma but not Elias Delta. I think, there was a problem with the number of zeros. On the other hand, in Huffman coding, they told us that we should follow the alphabetical order while constructing the tree, but they did not while encoding. Also, choosing the nodes when there is equality was unclear for some situations.

Self-criticism: We did not include the end character in our alphabet.

2) Group 1's stream of numbers: -6, 3, 4, -1

Their encoded message using Elias Gamma:

001100011100100110 (red-> number, blue-> +/-)

Encoded message if unary coding was used:

0111111001111011111010

For this case, the difference is only 4 bits. However, if the numbers were larger and there were more numbers than 4, unary coding would be inefficient.

- 3) In alphabet 1, we have {a, b, c, d, e, f, g, h, i, _, 'end'} as 11 characters. Therefore, 2^4 (4 bits) would be enough to encode it in binary format. (Even if we do not count space and end characters, 4 bits is needed.)

a	0000
b	0001
c	0010
d	0011
e	0100
f	0101
g	0110
h	0111
i	1000
_	1001
'end'	1010

Huffman coding according to Group 2's info:

a	001
b	10
c	00000
d	00001
e	11
f	0100
g	0001
h	011
i	0101
_	-
'end'	-

The expected code length for Huffman encoding in our case is

$$\sum p_i \cdot l_i = (5 \cdot 3 + 2 \cdot 7 + 5 \cdot 1 + 5 \cdot 1 + 9 \cdot 2 + 3 \cdot 4 + 2 \cdot 4 + 5 \cdot 3 + 3 \cdot 4) / 30 = \mathbf{3.2}$$

Fixed \rightarrow 4 bits, Huffman \rightarrow 3.2 bits

Huffman coding is more efficient than fixed-length coding.

- 4) As we discussed during the seminar, a better way to do arithmetic encoding could be to order the characters from the most frequent to the least frequent. This reduces the search for the correct interval while decoding.
- 3) I was happy to work with Group 3. Felicia and I were more involved in the computational part of arithmetic encoding, we computed the intervals individually and then compared them to make sure that there is no mistake. For the decoding part, we worked equally.