Source Coding in Brief

NOTE:

A source code should be decodable and has the least value of L (or the highest efficiency). This is achieved by good code design by providing the following:

A. A decodable code should have prefix property. Prefix property means any codeword should not be a prefix of any other codeword. (as in Code#4).

B. To have small L, one should look at the equation: L = $\sum_x P(x_i) \, . \, l_i$ and deduced that we must choose small l_i for large $P(x_i)$.

3- Source Coding Design

The word design means that we have the source with symbol probabilities, construct or find the code table with least L and should be decodable.

A. Design of Fixed Length Code:

This is very simple and we just need to find L. Using $L = \lceil Log_D M \rceil$ where M is the number of source symbols, and D is the code alphabet size (for binary code D=2).

The operator $\lceil \quad \rceil$ means the upper integer (also called the ceiling).

The code assignment may take any value.

Example-1 Design fixed length binary code for the source shown below:

Symbol x_i	Probability $P(x_i)$	Fixed Length Code	
		Codeword	l_i
x_1	0.05	000	3
x_2	0.2	110	3
x_3	0.1	010	3
x_4	0.3	111	3
x_5	0.15	011	3
x_6	0.08	101	3
x_7	0.12	100	3
$L = \lceil Log_D M \rceil = \lceil Log_2 7 \rceil = \lceil 2.807 \rceil = 3$		L= 3 Bits/Symbol	
$\eta_{sc} = \frac{H(x)}{L}.100\% = ?$		$\eta_{sc} = ?$	

Example-2 Repeat Example-1 for Ternary code 0,1,2 (D=3)

Symbol x_i	Probability $P(x_i)$	Fixed Length Code	
		Codeword	l_i
x_1	0.05	00	2
x_2	0.2	10	2
x_3	0.1	21	2
x_4	0.3	02	2
x_5	0.15	11	2
x_6	0.08	20	2
x_7	0.12	22	2
$L = \lceil Log_D M \rceil = \lceil Log_3 7 \rceil = \lceil 1.771 \rceil = 2$		L= 2 Ternary unit /Symbol	
$\eta_{sc} = \frac{H(x)}{L}.100\% = ?$		$\eta_{sc} = ?$	

B. Design of Variable Length Code:

Method#1: Fano Method

In this method the code construction is performed bit by bit to ensure decodable and least length code.

Design Steps:

1- Arrange all source symbols in descending order according to their probabilities.
2- Divide the symbols into D subsets, with almost equal sum probabilities in subsets.
3- Assign different code alphabet to each subset.
4- Repeat steps-2 and 3, until there is only one symbol in the subset.

Example-3 Design binary Fano code for the source in Example-1 (D=2)

Symbol x_i	Probability $P(x_i)$	1^{st} Bit Assignment	2^{nd} Bit Assignment	3^{rd} Bit Assignment	4^{th} Bit Assignment	Assigned Codeword	l_i
x_4	0.3	0	0			00	2
x_2	0.2	0	1			01	2
x_5	0.15	1	0	0		100	3
x_7	0.12	1	0	1		101	3
x_3	0.1	1	1	0		110	3
x_6	0.08	1	1	1	0	1110	4
x_1	0.05	1	1	1	1	1111	4

L = $\sum_x P(x_i) \, . \, l_i$= 2.63 Bits/Symbol H(x) = -$\sum_x P(x_i) \, . \, Log\, P(x_i) = 2.60288$ Bits/Symbol

$\eta_{sc} = \frac{H(x)}{L}.100\% = 98.97\%$ The above code is decodable.

Q.2 Repeat Example-3 with D=3.

Method#2: Huffman Method (Compact Coding)

In this method the code construction is performed in groups of D symbols to ensure minimum length and decodable code. In general, $L_{Huffman} \leq L_{Fano}$

Design Steps:

1- Arrange all source symbols in descending order according to their probabilities.

2- Sum the probabilities of the last D symbols, consider them as one symbol then re-arrange the symbols in descending order according to their probabilities.

3- Repeat step-2 until the final symbol (where the sum =1)

4- For each summing node, assign different code alphabet symmetrically.

5- The codeword for each symbol is given by those code alphabets assigned earlier along the traced path from the final node to given symbol.

Example-4 Design binary Huffman code for the source in Example-1 (D=2)

Q.3 Repeat Example-4 with D=3.

Example. (Huffman Code)