EE634: Information Theory A.Balaji
Date: 8-2-2005 EE04M004

Project #1: Optimal source code design for an *iid* binary source

Project Description:

To design an optimal source code for an iid source that outputs a 0 with probability 0.95 and a 1 with probability 0.05.

Algorithm:

1. The optimal source code for the given i.i.d source is the Huffman code.
2. Get the number of bits that must be considered together, from the user.
3. Generate all possible sequences of length that was obtained as input.
4. Also create the probability of each symbol.
5. For Ex.
   If there are 'm' zeros & 'n' ones in a symbol then its probability is (.95)^m * (0.05)^n.
6. Sort this table of symbols & their corresponding probabilities in decreasing value of probability.
7. Follow the Huffman coding procedure now. Since the symbols are sorted in decreasing probability order, first combine the first two probabilities and now sort this set of probabilities in decreasing order and continue this procedure till there are only two probabilities. An index table is created as we do the combining operation in each stage.
8. This indexing operation is done as follows:
   For example, there is a sorted list of probability values.consider 8 values, indexed as 1-8. Now if the first two elements are combined then there would be 7 values,and we do a new indexing from 1:7 and this is sorted now to get a jumbled set of indices. i.e the indices may not be in proper order from 1-7. This jumbled set is stored for each stage till the last stage.
9. After the indexing table is created. Now the code assignment has to be done. Assign codewords '0' and '1' to the last two probabilities. In the last stage, the indices will be 1 and 2. So 1 will correspond to the combined probabilities and so for whatever the codeword assigned for index 1, a '0' is appended to one of the probabilities and a '1' is appended to the other. And like this we move on from the last stage to the first stage, assigning codewords everytime. And so after reaching the last stage, we get the codewords for the symbol

Result:

Expected Length can be calculated using the formula

$$E(l(x)) = \sum_{x=1}^{M} p(x)\, l(x) \quad \text{where M= Total Number of symbols}$$

Expected Length per source bits = $E(l(x)) / \log_2 M$

For number of bits = 7, the expected length per source bit L is 0.3037 and marks is found using the equation $\frac{1-L}{1-h(0.95,0.05)}$ and is found to be 9.7570.

Source Code Table :

SourceWord	CodeWord	SoruceWord	CodeWord
1111111	1000011101010100010111111	1001100	1000011101000
1111110	1000011101010100010111110	0101100	1000011011101
1111101	1000011101010100010111001	0011100	1000011011100
1111011	1000011101010100010111000	1001010	1000011011111
1110111	1000011101010100010111011	0101010	1000011011110
1101111	1000011101010100010111010	0011010	1000011011001
1011111	1000011101010100010111101	1000110	1000011011000
0111111	1000011101010100010111100	0100110	1000011011011
1001111	1000011101010100001010	0010110	1000011011010
0101111	1000011101010100001001	0001110	1000011100101
0011111	1000011101010100001000	1001001	1000011100100
1111100	10000111010101000111	0101001	1000011100111
1111010	10000111010101000110	0011001	1000011100110
1110110	10000111010101001101	1000101	1000011100001
1101110	10000111010101001100	0100101	1000011100000
1011110	10000111010101001111	0010101	1000011100011
0111110	10000111010101001110	0001101	1000011100010
1111001	10000111010101001001	1000011	100001111101
1110101	10000111010101001000	0100011	100001111100
1101101	10000111010101001011	0010011	100001111111
1011101	10000111010101001010	0001011	100001111110
0111101	1000011101010100000101	0000111	100001111001
1110011	1000011101010100000100	1110000	100001111000
1101011	1000011101010100000111	1101000	100001111011
1011011	1000011101010100000110	1011000	100001111010
0111011	1000011101010100000001	0111000	1000011010101
1100111	1000011101010100000000	1100100	1000011010100
1010111	1000011101010100000011	1010100	1000011010111
0110111	1000011101010100000010	0110100	1000011010110
1001110	1000011101010100011	1100010	1000011010001
0101110	1000011101010100010	1010010	1000011010000
0011110	1000011101010111	0110010	1000011010011
1111000	1000011101010110	1100001	1000011010010
1110100	1000011101011101	1010001	100001110111
1101100	1000011101011100	0110001	100001110110
1011100	1000011101011111	1001000	100001100
0111100	1000011101011110	0101000	100001001
1110010	1000011101011001	0011000	100001000
1101010	1000011101011000	1000100	100001011
1011010	1000011101011011	0100100	100001010
0111010	1000011101011010	0010100	100000101
1100110	10000111010010101	0001100	100000100
1010110	10000111010010100	1000010	100000111
0110110	10000111010010111	0100010	100000110
1110001	10000111010010110	0010010	100000001
1101001	10000111010010001	0001010	100000000
1011001	10000111010010000	0000110	100000011
0111001	10000111010010011	1000001	100000010
1100101	10000111010010010	0100001	10001101
1010101	10000111010011101	0010001	10001100
0110101	10000111010011100	0001001	10001111
1001101	10000111010011111	0000011	10001110
0101101	10000111010011110	1100000	10001001
0011101	10000111010011001	1010000	10001000
1100011	10000111010011000	0110000	10001011
1010011	10000111010011011	0000101	10001010
0110011	10000111010011010	1000000	1011
1001011	10000111010100101	0100000	1010
0101011	10000111010100100	0010000	1101
0011011	10000111010100111	0001000	1100
1000111	10000111010100110	0000010	1111
0100111	10000111010100001	0000001	1110
0010111	10000111010100000	0000100	1001
0001111	1000011101010101	0000000	0