Information Theory: Assignment #3

Due on February 12, 2014 at 3:10pm

Marek Smieja

Szymon Maszke

Problem 1

Prove that Shannon coding is prefix.

Solution

Proof. A b c

Problem 2

Let $A = \{0, 1\}$ be the source alphabet and $P = \{0.1, 0.9\}$ it's probability distribution.

- Find Huffman and Shannon codes for all pairs, triplets and quadruplets of source alphabet.
- Compare their estimated code lengths attributable on symbol with entropy source
- Consider distribution $Q = 0.4, 0.6$ and repeat the task

Solution

Part A

Let's calculate probabilties for each pair and triplet

- pairs: $[00, 01, 10, 11]$ described by probability vector: $[0.01, 0.09, 0.09, 0.81]$.
- triplets: $[000, 001, 010, 011, 100, 101, 110, 111]$ described by probability vector: $[0.001, 0.009, 0.009, 0.081, 0.009, 0.081, 0.081, 0.72]$

Part B

Sort symbols in decreasing order based on their probability, calculate length of each code using the formula:

$$l_i = \lceil -\log_2(p_i) \rceil \tag{1}$$

and calculate their cumulative probabilities using formula:

$$P_c = \sum_{i<k} p(i) \tag{2}$$

- $[11, 10, 01, 00]$ with $p_p = [0.81, 0.09, 0.09, 0.01]$, $l_p = [1, 4, 4, 7]$ and $P_{cp} = [0, 0.81, 0.90, 0.99]$
- Analogous for triplets...

To receive the code we have to take first l_k digits from cumulative probability transformed to binary digits, so we receive:

- Binary cumulative probability: $P_{cpb} = [0.0, 0.1100, 0.1110, 0.1111111]$, which gives us the code: $[11 : 0, 10 : 1100, 01 : 1110, 00 : 1111111]$
- Analogous for triplets...

Source entropy of pairs is given by:

$$\sum_i p_i \log_2 p_i \leq h(X) + 1 \tag{3}$$

Average code length for Shannon coding is given by:

$$\sum_i p_i \lceil \log_2 p_i \rceil \leq h(X) + 1 \tag{4}$$

, with the sum being equal to element-wise vector multiplication of l_p and p_p.
Applying above formulas for pairs gives us:

$$0.81 + 2 * 0.36 + 0.07 < 0.81 * 0.246 + 2 * 0.312 + 0.066 + 1 \tag{5}$$

$$1.6 < 0.88926 + 1 \tag{6}$$

which is consistent with the approximation for average length of Shannon coding.

Problem 2 continued on next page...

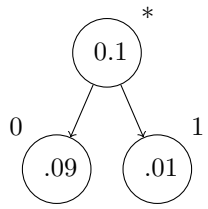
Part C

Introduction:
Huffman coding is optimal based on average code length (in contrast to Shannon coding). Based on binary trees with leaves representing symbols and and path from the root to leaves their codes (respectively).

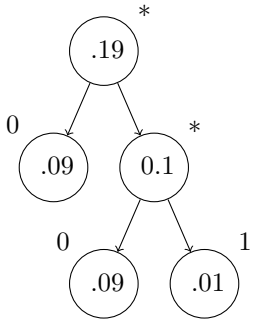
Solution

To create Huffman code we have to use recursive approach for binary tree creation. Starting with two smallest probabilities and differentiating between them only with one value, we climb up the probability 'ladder'. For each pair, we sum their probability and apply it back to the list.

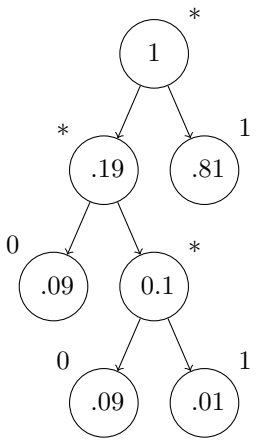
Our sorted probability list for pairs is: $p_p = [0.81, 0.09, 0.09, 0.01]$ and appropriate symbols are: $SYMBOLS = [11, 01, 10, 00]$.
We take two smallest values to create first binary tree. It will be parametrized by probability and appropriate code (root will be given code placeholder: *)



Our current list consists of following probabilities: $p_p = [0.81, 0.10, 0.9]$ and unified symbols:
$SYMBOLS = [11, *, 01]$, where $*$ is unified 00 and 10



Our current list consists of following probabilities: $p_p = [0.81, 0.19]$ and unified symbols:
$SYMBOLS = [11, *]$



Probability equal to one finishes the algorithm and we can code each symbol (moving to the left applies zero, while moving to the right applies one):

$$[11, 01, 10, 00] = [1, 00, 010, 011] \tag{7}$$

Based on the coding above $l_p = [1, 2, 3, 3]$ and probabilities for each code is respectively: $p_p = [0.81, 0.09, 0.09, 0.01]$, so average code length is:

$$0.81 + 5 * 0.09 + +0.01 * 3 \leq h(X) + 1 \tag{8}$$

$$1.29 \leq 0.88926 + 1 \tag{9}$$

It should be noted, that average code length for Huffman coding is smaller than the one found using Shannon coding with 1.29 and 1.6 respectively, hence can be considered more optimal with respect to this attribute.

FINISH OTHER EXAMPLES SOMEDAY...

Problem 3

Code text 'alabla' using arithmetic coding. Estimate probability distribution from data.

Solution

Part A

First we have to estimate probability distribution of a given string, as the task is trivial I will only provide a dictionary containing it:

$$[a : 0.5, l : 0.(3), b : 0.1(6)] \tag{10}$$

Part B

To code the string we have to use cumulative probability, e.g.:

$$\sum_{i<k} p_i = [0, 0.5, 0.83, 1] \tag{11}$$

- We start by coding first letter 'a' giving it $I_0 = [0, 0.5]$
- For letter l: $I_1 = 0 + 0.5 * [0.5, 0.83] = [0.25, 0.415]$
- For letter a: $I_2 = 0.25 + 0.165 * [0, 0.5] = [0.25, 0.3325]$
- For letter b: $I_3 = 0.25 + 0.08325 * [0.83, 1] = [0.3190975, 0.33325]$
- For letter l: $I_4 = 0.3190975 + 0.0141525 * [0.5, 0.83] = [0.32617375, 0.330844075]$
- For letter a: $I_5 = 0.3261737 + 0.004670325 * [0, 0.5] = [0.32617375, 0.3285088625]$

Result of this coding is the interval $[0.32617375, 0.3285088625]$, the marker can be given by it's mean: $\mathbf{z = 0.32734130625}$ and length of the word **n=6**

Part C

Add decoding if you have time

Problem 18