Question #1 : Proof of Correctness of Huffman Encoding Problem

Allen Kim, Yeshuchan (Jack), Gautam Ramasubramanian

October 16, 2016

Problem Statement

Justify the correctness of the Huffman encoding algorithm: In Huffman Encoding Algorithm (sort the symbols in order and merge the top two symbols iteratively until only one node is left), prove that for two symbols A and B with probabilities,

$$p(A) \geq p(B)$$

then in the result representation sequence according to the Huffman Encoding Procedure, the length of symbol A is no longer than that of symbol B. In other words, if L is a function the outputs the length of a symbol, then

$$H(A) \leq H(B)$$

Huffman Encoding Intro

Suppose we want to encode the following set of symbols - They could be ascii characters, unicode characters, etc. We will denote them as

$$V = v_1, v_2, v_3 \dots v_n$$

We have a file that contains all these symbols, but the frequency with which they appear on the file is different. In fact, let us assume that the following is true.

$$f_1 \leq f_2 \leq f_3 \dots f_n$$

In other words, v_n is the most frequent character in the file we want to encode, and v_1 is the least frequeny character in the file.

The Huffman encoding algorithm works as follows. We arrange the symbols in a queue (which is FIFO) in order of least to greatest frequency.

$$Q = [v_1, v_2 \dots v_n]$$

We pop the first two symbols out and merge them into one symbol. In this case we pop v_1 and v_2 and merge them to create $v_{1,2}$. The frequency of this merged symbol is the sum of the frequencies of the individual symbols.

$$f_{1,2} = f_1 + f_2$$

We input this merged symbol to the back of the queue, and we sort the queue from least frequency character to greatest frequency character.

Then, we iterate the procedure over and over again, removing the first two symbols, merging them, adding them back to the queue, and sorting the queue. We continue until there is only one symbol remaining in the queue, which is the merger of all the individual symbols.

The Huffman Length, denoted by the function L, is the number of times a symbol has been merged throughout the course of the algorithm. We want to prove that

$$L(v_i) \geq L(v_j) \qquad (i < j)$$

for all i and j. For merged symbol, L is the number of times a symbol has been merged after the merged symbol was created. Therefore,

$$L(v_{1,2}) = L(v_1) - 1 = L(v_2) - 1$$

1 Motivation for Proof

Lemma 1. Suppose we have a set of n symbols, $v_1, v_2, \ldots v_n$ with certain frequencies $f_1, f_2 \ldots f_n$. Let us assume that, without loss of generality, that $f_1 \leq f_2 \leq f_3 \ldots f_n$. Now, let us add an additional property.

Assume that

$$f_1 + f_2 \geq f_n$$

Then $L(v_1) \geq L(v_2) \geq \ldots \geq L(v_n)$. Furthermore, $L(v_1) - L(v_n) \leq 1$.

Proof. The proof is by induction. For the base case, suppose then there are three symbols v_1, v_2 and v_3. Then, v_1 and v_2 will be merged together to get $v_{1,2}$, and that in turn will be merged with v_3 to get $v_{3,1,2}$. Therefore, $L(v_1) = L(v_2) = 2$ and $L(v_3) = 1$. This satisfies the inequality $L(v_1) \geq L(v_2) \geq L(v_3)$ and $L(v_1) - L(v_3) = 1 \leq 1$.

will be the end of the Huffman encoding process. $L(v_1) = L(v_2) = 1$, so $L(v_1) \geq L(v_2)$ and $L(v_1) - L(v_2) = 0 \leq 1$. Now, suppose that the theorem were true for $n-1$ symbols. Now, we consider the case where there are n symbols in the queue.

$$Q = [v_1, v_2, \ldots, v_n]$$

In the first step of the Huffman Encoding problem, the first two symbols are merged together to get $v_{1,2}$ whose frequency is $f_{1,2} = f_1 + f_2$. Due to the property above, the resulting queue will look like this.

$$Q = [v_3, v_4, \ldots v_n, v_{1,2}]$$

Now, we can see that this is a queue with $n-1$ symbols. Furthermore, it is true that $f_3 + f_4 \geq f_{1,2} = f_1 + f_2$. So, if we make the replacement

$$\begin{aligned} w_1 &= v_3 & g_1 &= f_3 \\ w_2 &= v_4 & g_2 &= f_4 \\ &\vdots & & \\ w_{n-2} &= v_n & g_{n-2} &= f_n \\ w_{n-1} &= v_{1,2} & g_{n-1} &= f_{1,2} = f_1 + f_2 \end{aligned}$$

Then, we see that we have a set of $n-1$ symbols such that $w_1 + w_2 \geq w_{n-1}$, and whose frequencies satisfy $g_1 \leq g_2 \leq \vdots g_{n-1}$. Therefore, by induction it must be true that

$$L(w_1) \geq L(w_2) \ldots L(w_{n-2}) \geq \vdots L(w_{n-1})$$

Substituting the v terms back, we get

$$L(v_3) \geq L(v_4) \ldots L(v_n) \geq L(v_{1,2})$$

However, we also know that $L(w_1) - L(w_n) \leq 1$. Therefore, $L(v_3) - L(v_{1,2}) \leq 1$. Here, there are two cases.

Case 1 $L(v_3) - L(v_{1,2}) = 1$. Therefore, $L(v_3) = L(v_{1,2}) + 1 = L(v_1) = L(v_2)$. Since $L(v_3) - L(v_{1,2}) = L(w_1) - L(w_{n-1}) = 1$, we can say that $L(w_1) - L(w_{n-2}) = L(v_3) - L(v_n) \leq 1$. Since $L(v_3) = L(v_1)$, then $L(1) - L(n) \leq 1$ and $L(v_1) \leq L(v_2) \leq \ldots \leq L(v_n)$.

Case 2 $L(v_3) - L(v_{1,2}) = 0$. Therefore, $L(v_3) + 1 = L(v_1) + L(v_2)$. Since $L(v_3) - L(v_{1,2}) = L(w_1) - L(w_{n-1}) = 0$, we can say that $L(w_1) - L(w_{n-2}) = L(v_3) - L(v_n) = 0$. Therefore, $L(v_1) - L(v_n) = 1 \leq 1$, and $L(v_1) \leq L(v_2) \leq \ldots \leq L(v_n)$. □