Jared Nielson
u0495206
Assignment 12 Analysis
CS 2420

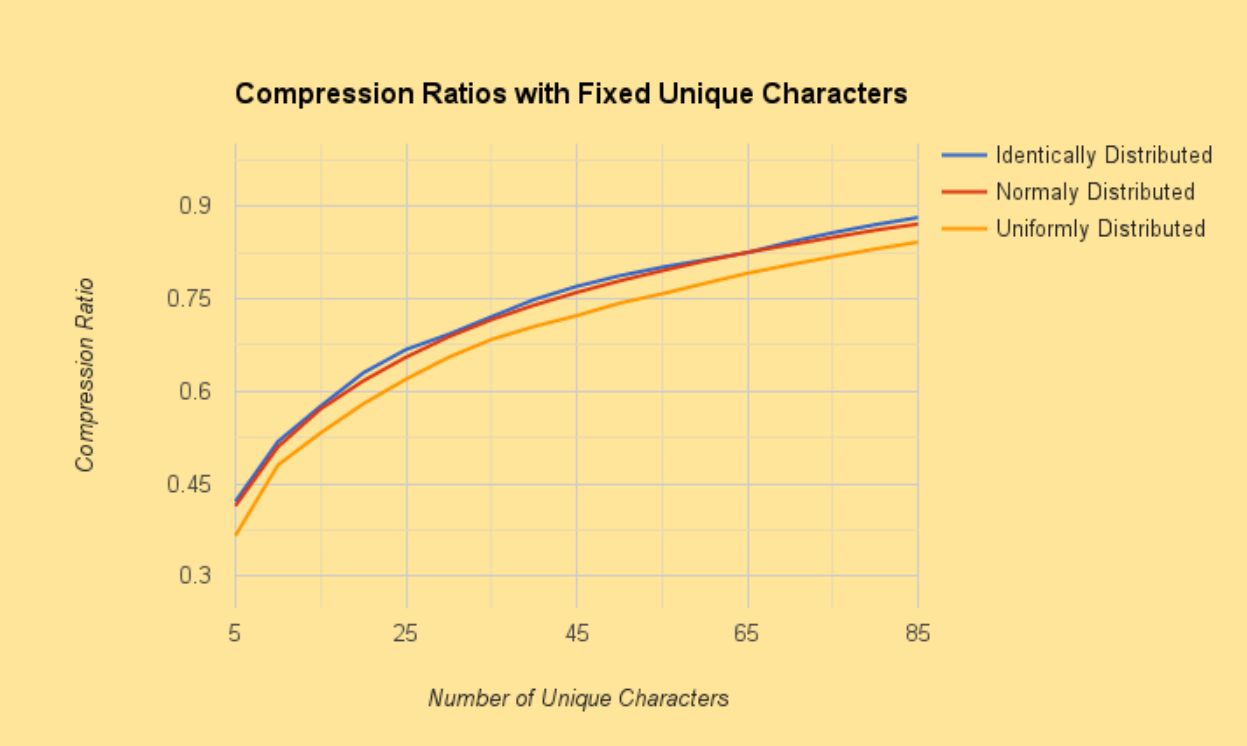
1. **Design and conduct an experiment to evaluate the effectiveness of Huffman's algorithm. How is the compression ratio (compressed size / uncompressed size) affected by the number of unique characters in the original file and the frequency of the characters?**

To test the compression ratio based on unique characters the following experiment was designed.

1. A text file was created with a fixed number of unique characters (5). The frequency of these characters were all held constant at 75 occurrences/character.
2. The file was compressed.
3. The compression ratio was determined by dividing the compressed files size by the original file's size.
4. The number of unique characters was increased by 5 and the process was repeated until the number of unique characters was 85.
5. Steps 1-4 were repeated using a random frequency with uniform distribution and an average number of occurrences of 75.
6. Steps 1-4 were repeated using a random frequency with a normal distribution with a mean of 75 and a standard deviation of 15.

The results of the experiment are posted below. As the number of unique characters increases so does the compression ratio. The best compression ratios are achieved with the lowest number of unique characters. This is expected as the binary trie will have fewer levels for a lower number of unique characters.

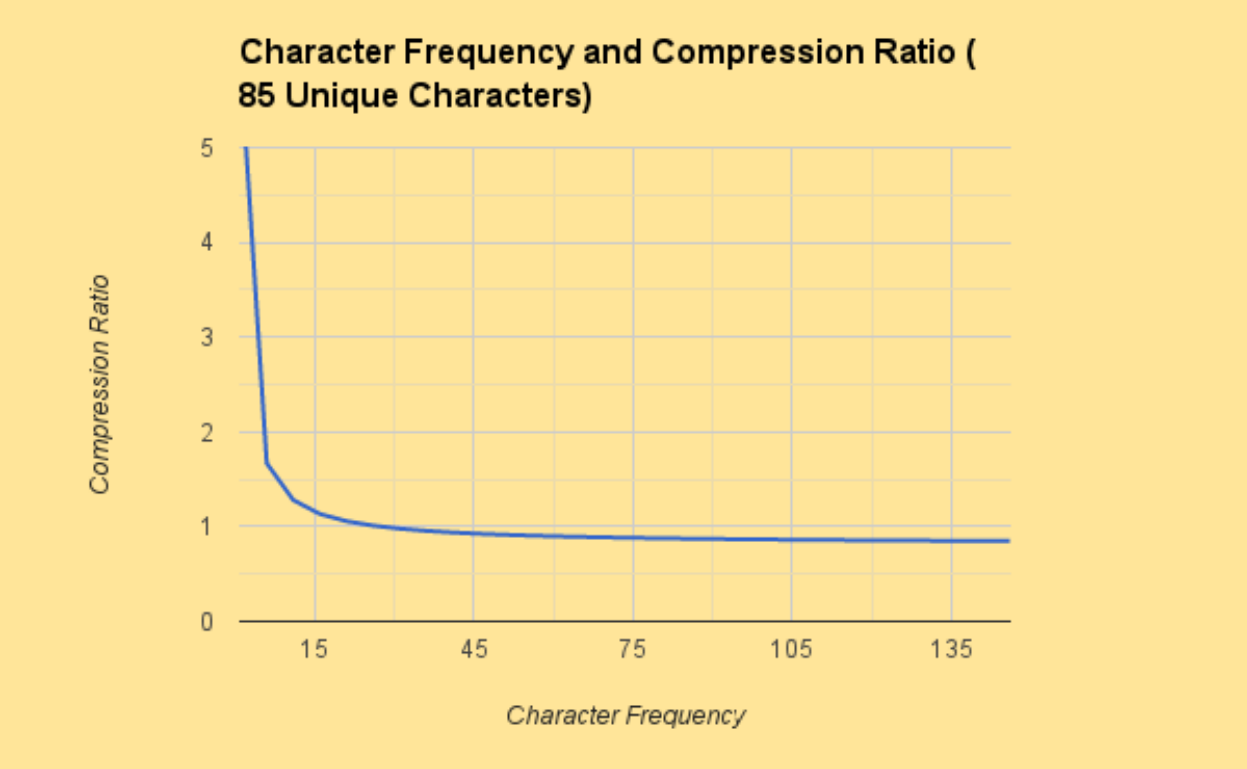
Text files that have a normal distribution had the best compression ratio. This is also expected as there are a fewer number of characters with a higher frequency that get placed at higher levels of the binary trie as opposed to most characters being at the same depth of the binary trie with the uniform distribution.



To test the compression ratio compared to frequency the following experiment was performed.

1. A text file with 85 unique characters was generated. Each character in the file had a frequency of 1.
2. The file was compressed and the compression ratio was determined.
3. Steps 1-2 were repeated after increasing the frequency by 5 until a frequency of 150 was reached.

The results of the experiment are pasted below.



As overall character frequency increases so does the compression ratio. In fact for very low frequencies the compressed file was actually larger than the original file. This is expected as the overhead for storing the compression information will be larger than the space saved for a low frequency of characters.

2. For what input files will using Huffman's algorithm result in a significantly reduced number of bits in the compressed file? For what input files can you expect little or no savings?

As shown in the experiments above Huffman Compression works much better for files with a lower number of unique characters that each have a high frequency. However most text files of sufficient length will benefit from Huffman Compression.

It is clear from the second experiment that files with many unique characters that each only occur a few times, less than 20, will have no savings, or even an increased cost.

3. Why does Huffman's algorithm repeatedly merge the two smallest-weight trees, rather than the two largest-weight trees?

By merging the smallest weighted trees first it guarantees that characters with the highest frequency occur closer to the root (at higher levels of the tree). This is important because the closer a character is to the root node the fewer bits it will take to encode that character. In order to get a good compression the most frequently used characters should require the fewest number of bits to encode. Creating the tree by merging the smallest-weight trees accomplishes this.

4. Does Huffman's algorithm perform lossless or lossy data compression? Explain your answer.

Huffman's algorithm provides a lossless compression. This means that the data that is compressed can be decompressed without losing any information. When a file is compressed and decompressed that decompressed file is identical to the original file. There is no data loss.

5. How many hours did you spend on this assignment?

I spent approximately 5 hours on this assignment.