A Study of Different Data Compression Techniques

Shashank Gautam*

Department of E.C.E., R.B.S. Engineering Technical Campus, Bichpuri-Agra, India

(shashankgautam92@gmail.com)

***Abstract*- Data compression refers to reducing the volume of space needed to keep data or reducing the amount of time required to transmit the data. The size of data is diminished by removing the unnecessary information. In this paper many different data compression techniques have been studied such as Shannon Fano, Shannon Fano Elias, Huffman binary and ternary coding. It is found that Huffman coding is most suitable from the among techniques as it provides smallest codeword length.**

Keywords— Data Compression; Shannon Fano; Shannon Fano Elias; Huffman Binary Coding; Huffman Ternary Coding

1. Introduction

Data compression refers to reducing the volume of space needed to keep data or reducing the amount of time required to transmit the data. The size of data is diminished by removing the unnecessary information. It is a process by which any type of data i.e. Text, Audio or Video can be altered to compressed file, such that the original file can be fully recovered from the compressed file deprived of any loss of actual information. This process is useful to save the storage space. In other words it can be said that data compression is the process of encoding the data to fewer bits than the original representation so that it occupies less storage space and lesser amount of transmission time during communicating over a network [1].

Data Compression strive to decrease the amount of bits used to save or transmit the information in a frame. Compression is a process to reduce then the physical size of information but keeping its meaning. Compression techniques which are unlike to each other but have something in common that they all compress information bits [2].

2. Types of Data Compression

Till now two fundamental classes of data compression are associated in various areas. One of these is lossy data compression which normally uses to compress picture information documents for correspondence or files purposes. The other is lossless data compression that is regularly used to transmit or file content or parallel records needed to keep their data in place [3].

2.1. Lossless Compression

Decreases bits by recognizing and removing redundancy, none of the information is lost in lossless compression. In these techniques before the compression after the compression data must be similar.

2.2. Lossy Compression

Decreases bits by recognizing marginally important data or information and removing it. In these techniques some loss of data is tolerable depending upon the application.

3. Shannon Fano Coding

In Shannon Fano coding, the procedure is done by a more frequently following string which is encoded by a shorter encoding vector and fewer frequently occurring string is encoded by longer encoding vector. It is a technique for creating a prefix code centered on a set of symbols and their corresponding probabilities [4].

<table>
<tr><th>a_i</th><th>$p(a_i)$</th><th>1</th><th>2</th><th>3</th><th>4</th><th>Code</th></tr>
<tr><td>a_1</td><td>0.36</td><td rowspan="2">0</td><td colspan="3">00</td><td>00</td></tr>
<tr><td>a_2</td><td>0.18</td><td colspan="3">01</td><td>01</td></tr>
<tr><td>a_3</td><td>0.18</td><td rowspan="4">1</td><td colspan="3">10</td><td>10</td></tr>
<tr><td>a_4</td><td>0.12</td><td rowspan="3">11</td><td colspan="2">110</td><td>110</td></tr>
<tr><td>a_5</td><td>0.09</td><td rowspan="2">111</td><td>1110</td><td>1110</td></tr>
<tr><td>a_6</td><td>0.07</td><td>1111</td><td>1111</td></tr>
</table>

Figure 1. Code generation by Shannon Fano Coding.

3.1. Algorithm of Shannon Fano Coding

Step 1: Create table providing probabilities counts.

Step 2: Sort symbols according to their corressponding probabilities in descending order.

Step 3: Recursively divided the given set of probabilities into two parts, each with approximately same number of counts.

Step 4: Add a binary 1 to the code words of the lower part and a binary 0 to the upper part.

Step 5: Search for the next part containing more than two symbols and repeat the step 3 and step 4 to obtain codewords.

Figure 1 shows the generation of codewords using the above mentioned algorithm of Shannon Fano coding.

4. Shannon Fano Elias Coding

In information theory and coding , Shannon Fano–Elias coding is a pioneer to arithmetic coding, in which probabilities are used to determine the codewords.[5]

4.1. Algorithm of Shannon Fano Elias Coding

Step 1: Create table providing probabilities counts.

Step 2: Sort symbols according to their corressponding probabilities in descending order.

Step 3: Cumulative distribution function F(x) is calculated .

$$F(x) = \sum_{a \le x} p(a)$$

Step 4: $\bar{F}(x)$ is calculated which denotes the sum of the probabilities of all symbols less than x plus half the probability of the symbol x.

$$\bar{F}(x) = \sum_{a<x} p(a) + \frac{1}{2} p(x)$$

Step 5: $\bar{F}(x)$ is converted into binary and is noted down.

Step 6: Expected length of code i.e. l(x) is calculated.

$$l(x) = \lceil \log \tfrac{1}{p(x)} \rceil + 1$$

Step 7: Finaly the corresponding codewords are obtained. Table 1. shows an example encoding a setof data using Shannon Fano Elias coding procedure.

Table 1. Shannon Fano Elias Coding.

x	p (x)	F (x)	$\bar{F}$ (x)	$\bar{F}$ (x) in binary	l (x)	Codewords
1	0.250	0.25	0.125	0.001	3	001
2	0.050	0.75	0.5	0.10	2	10
3	0.125	0.875	0.8125	0.1101	4	1101
4	0.125	1.0	0.9375	0.1111	4	1111

5. Huffman Coding

The Huffman code is a source code, in this the word length of the code word approaches the fundamental limit set by the entropy of discrete memoryless source. The codeword obtained by Huffman coding is said to be optimum as it provides the smallest average codeword length for a given memoryless source. Huffman coding is basically of two types i.e. Huffman binary code and Huffman ternary code [5,6].

5.1. Algorithm of Huffman Binary Coding

Step 1: List all the source symbols i.e. messages in the order of decreasing probabilities.

Step 2: The two source symbols of lowest probability are assigned numbers 0 and 1. This is reffered to as splitting stage.

Step 3: The two source symbols are combined into a new message. The probability of this new message is equal to the sum of probabilities of the two original symbols.

Step 4: The probability of the new symbol is placed in the list of symbols in accordance with its value.

Step 5: The procedure is repeated until we are left with only two source symbols for which a 0 and a 1 are assigned.

Step 6: The code of each original source symbol is obtained by working backward and tracing the seuence of 0s and 1s assigned to that symbol as well as its successors.

Figure 2 shows the generation of codewords using the above mentioned algorithm of Huffman binary coding.

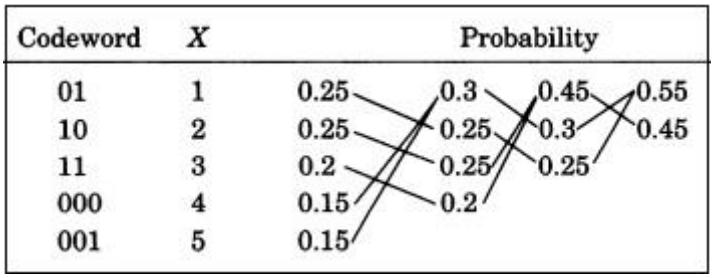

Figure 2. Code generation by Huffman Binary Coding.

5.2. Algorithm of Huffman Ternary Coding

Step 1: List all the source symbols i.e. messages in the order of decreasing probabilities. If there are even number of source symbols then a dummy symbol is added in the list with probability 0.

Step 2: The three source symbols with lowest probability are assigned numbers 0, 1 and 2.

Step 3: The three source symbols are combined into a new message. The probability of this new message is equal to the sum of probabilities of the three original symbols.

Step 4: The probability of the new symbol is placed in the list of symbols in accordance with its value.

Step 5: The procedure is repeated until we are left with only three source symbols for which a 0, a 1 and a 2 are assigned.

Step 6: The code of each original source symbol is obtained by working backward and tracing the seuence of 0s, 1s and 2s assigned to that symbol as well as its successors.

Figure 3 shows the generation of codewords using the above mentioned algorithm without dummy symbol.

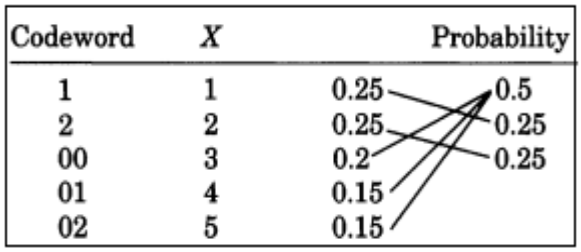

Figure 3. Code generation by Huffman Binary Coding without dummy variable.

Figure 4 shows the generation of codewords using Huffman ternary coding algorithm with dummy symbol.

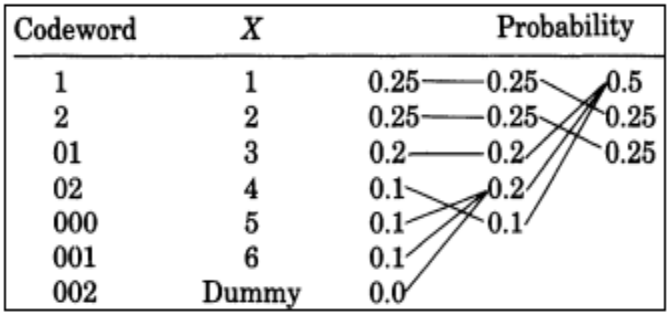

Figure 4. Code generation by Huffman Binary Coding with dummy variable.

6. Conclusion

The paper presents different techniques for data compression along with their algorithms. Different data compression techniques have been studied and it is found that Huffman coding is most suitable for data compression as it provides the smallest average codeword length.

References

[1] S. Porwal, Y Chaudhary, J. Joshi and M. Jain, “Data Compression Methodologies for Lossless Data and Comparison between Algorithms,” International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 2, March 2013

[2] S.R. Kodituwakku and U. S. Amarasinghe , "Comparison Of Lossless Data Compression Algorithms For Text Data" Indian Journal of Computer Science and Engineering Vol1No. 4 416-425

[3] Senthil Shanmugasundaram, Robert Lourdusamy ”A Comparative Study Of Text Compression Algorithms”. International Journal of Wisdom Based Computing, Vol.1 (3), December 2011

[4] H.Altarawneh and M. Altarawneh, " Data Compression Techniques on Text Files: A Comparison Study ” International Journal of Computer Applications, Volume 26– No.5, July 2011

[5] T.M. Cover and J.A. Thomas, “Elements of Information Theory,” John Willey & Sons. ISBN 9788126541942.

[6] P. Yellamma, N. Challa, " Performance Analysis Of Different Data Compression Techniques On Text File" International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 8, October – 2012, pp.1-6