On Compression Function

Shashi Kant Pandey[1], Vijay Dahiya[2], Harish Singh[3]
Department of Business Administration,
Maharaja Surajmal Institute, GGSIP University
Delhi-110058
[1]shashikantpandey@msi-ggsip.org
[2]vijaydahiya@msi-ggsip.org
[3]harishsingh@msi-ggsip.org

Abstract— keeping the data with it long lasting validity is one of the primary requirements in computer science. The storage size of data and the speed of access plays interesting role in this procedure. There are various storage schemes available in literature. Development of a scheme is always an attractive field of research for commuter scientist. The length of the code plays an important role in designing of the scheme. In this article we discuss a well-known scheme namely Huffman coding scheme and its algorithm with an example. A brief history and current trends of compression functions are also discussed.

Keywords— compression functions; coding theory; tree.

I. INTRODUCTION

The efficient transmission of data through online mode is very interesting task for network analyst and computer scientist. Here the meaning of efficiency of transmission mode is in two aspects, the accuracy of the received data and the time duration between sender and receiver. Through data compression it can achieve in some of aspects of these requirements. To fast transfer of data at its destination, it is necessary to either increase the data rate of the transmission media or simply send less data. But, we never require less data or a data with full of error at the receivers end. So always it is mandatory that either we received error free data or the whole data without any loss of information. Suppose we have to send a file through mail then first we compress it (using win zip), which reduces the size of the data and receiver get the original file without any loss of information. Win zip helps us in reduction of the size but the problem during transmission is still available. This can solve by the speed of transmission of the data (using high speed bandwidth). It is interesting that optical fiber transfer the data at the rate of speed of light in the glass. Nowadays, fiber optics is used to increase the transmission rate of the data. Using fiber optics communication, researchers at Bell labs have achieved over 100 Petabits per second kilometer speed [6]. Research in the direction of fiber optics is going on and NTT japan had demonstrated one of the fastest single fiber cables in 2012 [7]. Search of compression functions is another branch of work in the optimal data transmission. Claude Shannon was the person whose work on information theory is a breakthrough for this science [8]. Later he and his colleague Feno, developed a famous coding scheme named as Shanon-Fano coding.

In 1951, David Huffman a student of Fano, produce an algorithm which is an optimal coding scheme and it is known as Huffman coding scheme. This scheme is based on properties of tree in Graph theory. In a tree there is one and only one path between any two vertices, Huffman used this concept and produce this optimal scheme. One of the techniques to use it for storage in more optimal way is to compress the files. By taking advantage of redundancy or patterns, it may be able to "abbreviate" the contents in such a way to take up less space yet maintain the ability to reconstruct a full version of the original when needed. Such compression could be useful when trying to cram more information on a disk or to shorten the time needed to copy or send a file over a network. There are some known compression algorithms, which give compression formats, such as JPEG, MPEG, or MP3, are specifically designed to handle a particular type of data file. Some of the compression algorithms (e.g. JPEG, MPEG) are lossy-decompressing. In this compression scheme, the compressed result doesn't recreate a perfect copy of the original. It has the algorithm which compresses by just summarizing the data. In process of reconstruction the summary losses some information and fail to reconstruct the original data. Sound and video data may be acceptable for lossy encoding schemes because in this case a huge amount of data is available. In case of some missing pixels, retrieving does not affect the original information. But for text data these lossy encoding schemes are not appropriate. In case of text data, whole information is distributed uniquely and in retrieving of the original text it is necessary to get each at most text correctly. Huffman scheme is an efficient scheme in this scene. It reduces the text loss in the retrieving process because it not works on the lossy method of compression. In this article we brief this technique with its mathematical background. Section 2 contains the requirement of concepts from graph theory. Further we elaborate the Huffman scheme with an example.

II. GRAPHS AND TREE

Graphs theory has remarkable analogy to solve a lot of mathematical problems. From the beginning to till date this area attracts the research community to explore the theoretical understating about graphs. In this section we present the essential theoretical details of graphs.

Definition 2.1 [Graph] A graph G is representation of the relation between two sets called vertices and edges. If E and V be the set of edges and vertices respectively then we denote the graph as $G = (V, E)$.

Now to understand the concept of tree in graph theory following definitions are essential here. For the sake of convenience we assumed that there are n number of vertices $\{v_i : 1 \leq j \leq n\}$ and m number of edges $\{e_i : 1 \leq i \leq m\}$ in a graph denoted as $G = (V, E)$.

Definition 2.2 [Path] Sequence of continuous vertices in a graph is called path. For example $v_1, v_2, \dots, v_k$ is a path of length of k.

Definition 2.3 [Connected Graph] A graph G is connected graph, if there is a path between any two vertices v_i and v_j of that graph for all $1 \leq i, j \leq$ n.

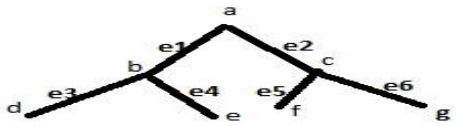
Definition 2.4[Walk] A walk in a graph is a path or a sequence of vertices with no repetition. A walk is closed if its start and end vertices is same.

Definition 2.5[Cycle] A closed walk is called a cycle in a graph. A graph which has no any cycle then it is called acyclic.

Based on the previous well known definitions in graph theory now we present a formal definition of tree in graph theory. Especially it is related to the unavailability of cycle in a graph and in the following definition we present this.

Definition 2.6[Tree] A connected acyclic graph is a tree. In a tree we always get a path from any two vertices and it is unique for any two vertices.

Here we present a graphical presentation of an example of tree of seven vertices $\{a, b, c, d, e, f, g\}$ and six edges $\{e1, e2, e3, e4, e5, e6\}$ is,



III. Primitives of Coding theory

Design of all compression functions are based on some techniques of a mathematical function which is one way. From starting to till date the designing of compression functions are an interesting are of research in cryptography and coding theory. Various security algorithms and cryptographic signature schemes are based on some efficient compression functions. Keccak is the latest design of a compression function. This is based on the technique of sponge function [9]. Here we present some theoretical aspects for the design for a compression function.

A. Perfect secrecy:

Achievement of perfect secrecy is always required in design of any cryptosystem. In practical scenario perfect secrecy means that Oscar can obtain no information about plaintext by observing the cipher text. This terminology can be described in terms of probability distribution also. It is defined as in following definition.

Definition 3.1 A cryptosystem has perfect secrecy if $Pr[x \mid y] = Pr[x]$ for all $x \in P, y \in C$. That is, the a posteriori probability that the plaintext is *x*, given that the cipher text y is observed, is identical to the a priori probability that the plaintext is *x*

B. Entropy:

The idea of entropy is introduced by Shannon at first time in his famous paper named "A Mathematical Theory of communication" in 1948 [8]. Entropy can be thought of as a mathematical measure of information or uncertainty, and is computed as a function of probability distribution.

Suppose we have probability distribution of some events. The amount of information we gain from occurrence of these particular events under given probability distribution or we say, at what extent the uncertainty about the outcomes of events which are not yet occurred. So from definition it is clear that entropy is a function of probability distribution. If X is a random variable then entropy of X is denoted by H(X).

Definition 3.2 Suppose $\boldsymbol{X}$ is a discrete random variable which takes on values from set X Then the entropy of the random variable $\boldsymbol{X}$ is defined to be the quantity

$$H(\boldsymbol{X}) = -\sum_{x \in X} pr[x] log_2 pr[x]$$

In the next section we show an example of compression function namely Huffman coding scheme.

IV. Huffman coding scheme example and Algorithm

This scheme is based on the probability distribution of the plaintext. Using the technique of graph theory and probability distribution, this scheme assigns a unique code to each of the plaintext. Following is the algorithm for the Huffman coding scheme.

A. Algorithm:

Let given probability distribution is $P = \{p_i : 1 \leq i \leq n\}$

Step1: Short the $p_i's$ in decreasing order such that $p_1 \leq p_2 \leq \cdots \leq p_n$ and assume they are vertices of tree. These vertices are called as leaf of this tree and we start the extension of that tree from its leaf to its root.

Step2: Choose two minimum of $\{p_i : 1 \leq i \leq n\}$, these are p_1 & p_2

Step3: Find new vertices with entry are sum of $\{p_1 \& p_2\}$ let it is q_1 & makes edge between$p_1 to\ q_1$ & $p_2\ to\ q_1$. In this way we can extend this tree.

Step 4: Make new set of probability distribution with removing two selected p_i and include the new vertices q_i in P.

Step 5: Do step2& step3 till the vertices end or when we rich at the root of the tree.

Step 6: For encoding start labeling the tree starting from root as left edge with 0 and right edge with 1.

Step 7: Choose path from the root to the pendent vertices to encode the character whose probability distribution is given. And in this way we can provide a unique code to every information very efficiently.

B. Example:

Huffman coding uses 'variable length coding' which means that symbols in the data which we want to encode are converted to a binary symbol based on how often that symbol is used.

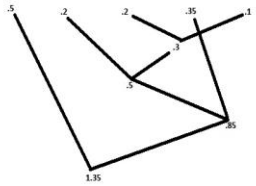
Let the probability distribution of five information or characters are as follows:

$$P = \{.5, .2, .2, .35, .1\}$$

First step: short P in descending order

$$P(x) = \{.5, .2, .2, .35, .1\}$$

Second step: Take smallest two of them and starting to build edge between both of them and sum of both probabilities. Again recursively do this process till all probability does not assign. Choose again minimum of them and add them for next vertices of the tree, we can see the minimum of them is .2 & .2 so next vertices is attached with them and its entry is .2+.2=.4 Recursive process of these steps gives a tree.



Give level to all edges of this tree which are code or alphabets. After leveling all the edges of this tree we get the required optimal code for this probability distribution of information.
Give level of this tree with 0 or 1 in the way such that all left edge are leveled with 0 and all right edges corresponding to each vertices are leveled with 1,we get the unique code for each characters. To produce code words for a character x follow the path from root of this tree to pendent vertices which is the probability corresponding to particular *x*. We get code words for each one as follows:

Cod words:	00	001	0101	11	1101
P(*x*):	.5	.2	.2	.35	.1

Since from each pair pendent vertices and root we have one and only one path so we get a unique code words for any " x"

Now we see that expected length of each code world is

l(*x*):	2	3	4	2	4
P(*x*):	.5	.2	.2	.35	.1
P(*x*)*l(*x*)	1.0	.6	.8	.7	.4

L =expected length of this code = $\sum P(x)*l(x)$ = 3.5 bits and we know that in binary value system the entropy of this P is $H_2(P)$= 2.5016 bits. Therefore we can see that $H \leq L \leq H+1$
So it satisfies source coding inequality and we can say that this scheme work like that the most probable information or character are represented by less bits and less probable character or information are represented by more bits so this algorithm can optimize the expected length and produce an optimal code.

C. Authors and Affiliations

1. Assistant Professor in Maharaja Surajmal Institute, C-4, janakpuri, Delhi 110058
2. Associate Professor in Maharaja Surajmal Institute, C-4, Janakpuri, Delhi-110068
3. Professor in Maharaja Surajmal Institute, C-4, Janakpuri, Delhi-110068

REFFRENCES

[1] Julie Zelenski with minor edits by Keith Schwarz "Hauffman Encoding and Data Compression", Spring 2012.
[2] J. Douglas R. Stinson "Cryptography Theory and Practice", 3rd edison, Champion and Hall/CRC, 9781-1-58488-508-5
[3] Ranjan Bose:Information Theory Coding and Cryptography ,T.M.H publication; page no 1-45.
[4] Wikipedia the encyclopedia.
[5] Harary :Graph Theory, Narosa Publishing House, 978-8185015552
[6] https://phys.org/news/2009-09-bell-labs-optical-transmission-petabit.html
[7] Chirgwin, Richard (Sep 23, 2012). "NTT demos petabit transmission on single fibre". The Register. Retrieved 2014-02-16.
[8] Shannon, C. E. (1938). "A Symbolic Analysis of Relay and Switching Circuits". *Trans. AIEE*. **57** (12): 713–723.
[9] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche Cryptographic spongeshttp://sponge.noekeon.org