A REVIEW OF LOSSLESS DATA COMPRESSION ALGORITHMS

A PREPRINT

Aditya Meharia
School of Computer Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, India 751024
adityameharia14@gmail.com

Junaid H Rahim
School of Computer Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, India 751024
junaidrahim5a@gmail.com

March 11, 2020

ABSTRACT

Objective

This work aims to provide an introduction to the domain of Data Compression in Information Theory and a review of the existing literature in the field of Algorithms for Lossless Data Compression. We identify and discuss the potential opportunities, barriers and the future scope of the field. We also review data compression methods used for text, image, video and audio data.

Results

Mostly about which algo performs best for what kinda field and all that. will write this later when the paper is complete. [1] thats a reference example

***K*eywords** Data Compression · Algorithms · Lossless Compression · Information Theory

1 Introduction

A rapid growth in modern communication technology led an explosion in the amount of data we transmit and store. Extremely large files consume significant resources for transmission as well as storage. Due to this exponential increase in the size of the data we transmit, researchers developed algorithms that can be used to compress the data to save storage space and transmission time. Data Compression is a process by which we encode the input data into a representation that occupies fewer bits than the original input. This encoded representation is transmitted and decoded back to the original form at the destination. Data compression algorithms are broadly classified into two classes viz **Lossless Compression** and **Lossy Compression** algorithms. We will be only covering Lossless Compression in this article.

Lossless Compression algorithms are a class of algorithms that can reproduce the original content from the encoded respresentation without any loss of information, the data before compression and after decompression is exactly the same. Lossless compression is used in a variety of fields where it is important that the original and decompressed information be the same. The GNU tool gzip uses lossless algorithms for the ZIP file format.

1.1 Overview of Lossless Data Compression Techniques

Lossless compression algorithms usually have a two step procedure.

1. A statistical model of the input data is generated. This usually assigns a probability of occurrence to pieces of input data. For example, if the input data is piece of text, then the model would be the probabilities of occurance of each alphabet
2. A coding system uses this model to map the data in a way that the pieces with high probability of occurrence are assigned a shorter code than those with a low probability of occurance

The probabalistic model is usually generated in two ways, a static way and an adaptive/dynamic way. In the static approach, the data is analysed and the probability model is generated before starting the compression, this is a modular and simple approach but doesn't perform well for heterogeneous data since the approach forces the use of a single model for the all the data. In the dynamic method the model is updated while compressing the data. The encoder and decoder start with a trivial model in the initial state, thus performs poorly on initial data, but as the model adapts to the data, the performance improves. Most efficient compression techniques usually employ an adaptive model.

2 Prefix Codes

3 Shanon Coding

It is named after its creator **Claude Shanon**, the technique was used to prove Shanon's noiseless coding theorem in his 1948 article "A Mathematical Theory of Communication". Even though being suboptimal, the method was a first of its kind. This method is credited to have given rise to the entire field of Information Theory. Some of the most efficient compression algorithms today are usually an exntension of shanon's method

Shanon Coding is a method to generate prefix codes for a given piece of data. It is done using the occurrence probabilities of the pieces of data. First the probabilities p_i are arranged in descending order, then each piece is assigned a code which is the first l_i digits of binary representation of the cumulative probability till that piece of data.

Given that the probability of occurrence is p_i, the cumulative probability is expressed as

$$\sum_{k=0}^{i-1} p_k$$

where $l_i = \lceil \log_2 p_i \rceil$

It is a suboptimal algorithm, it does not give the lowest possible code word length.

The following is an example of assigning prefix codes to compress the string "lossless data compression"

i	a_i	p_i	$p_c = \sum_{k=0}^{i-1} p_k$	Binary Representation	$l_i = \lceil \log_2 p_i \rceil$	code
0	s	0.24	0.00	0.00000000...	2	00
1	o	0.12	0.24	0.00111101...	3	001
2	e	0.08	0.36	0.01011100...	3	010
3	<space>	0.08	0.44	0.01110000...	3	011
4	a	0.08	0.52	0.10000101...	3	100
5	l	0.08	0.60	0.10011001...	3	100
6	i	0.04	0.68	0.10101110...	4	1010
7	d	0.04	0.72	0.10111000...	4	1011
8	t	0.04	0.76	0.11000010...	4	1100
9	c	0.04	0.80	0.11001100...	4	1100
10	m	0.04	0.84	0.11010111...	4	1101
11	r	0.04	0.88	0.11100001...	4	1110
12	p	0.04	0.92	0.11101011...	4	1110
13	n	0.04	0.96	0.11110101...	4	1111

4 Huffman Coding

5 Lempel-Ziv(lz) Compression Methods

6 Run Length Coding

7 Prediction by Partial Matching

8 Arithmetic Coding

9 Deflate

10 Grammar Based Compression

11 Current Research Work

12 Future Scope

13 Conclusion

JHR

References

[1] PM Parekar and SS Thakare. Lossless data compression algorithm–a review. *International Journal of Computer Science & Information Technologies*, 5(1), 2014.