Data Compression through Duplicate Elimination and Tagging

Jeffrey Thomas, 10/23/2008
Computer Systems Lab
First Quarter Research Paper

I. Abstract

Data compression is a valuable tool to save memory and time when sending data between computers. Although many different methods exist, I believe to have created a new method to compress data based on simple probability and the concept of effective infinite data. This algorithm will also potentially work with any kind of data, and will always compress a significant amount of data. I intend to test the effectiveness of this algorithm in terms of both time saved when sending large amounts of data and the proportion of data compressed, and find the optimal case of the inner variables.

II. Introduction

In order for a data compression algorithm to be valid for use, it must reach its optimal case a majority of time it is run on different data files. An algorithm is useless if it cannot reach its optimal case very often. Effective methods of data compression are vital in the current progress of computing. As file sizes grow exponentially, methods must be used to save space as well as time when sending data over networks.

In this paper, we will explore fully a completely original data compression method. This method relies on two inner variables, and can be run multiple times on compressed data. The first task will be to find the optimal state in one iteration, if one exists. It is possible that the algorithm will have no optimal state, as increasing the variables to infinity will infinitely enhance the performance. After the optimal state is found, the algorithm will be tested on multiple iterations with the optimal variables to find if there is an optimal number of iterations.

The algorithm will be tested on two major factors; the proportion of data compressed and the time saved when the compressed data is sent over a network, and then decompressed as opposed to sending it without any compression. The data files it sends will be randomly generated to simulate large files. In theory, the algorithm should be able to compress large data files to an optimal proportion any time it runs. This paper will test that claim.

If the algorithm is a success, it has the potential to be exceptionally useful in the world of computing, as it can perform on any type of file, and should be able to optimally perform every time it runs.

III. Background

The main purpose of data compression is to replace larger patterns of data with smaller representations. In doing so, there are two main methods; lossless, and lossy compression. Lossless compression does not lose any data in the compression/decompression process. It is used for data that requires accuracy, such as program and text files. Lossy compression allows for some data to be lost in the process, and is used in compressing images and other visual mediums (Data-Compression.com).

A common method of data compression is Huffman coding. This takes repeating symbols or patterns and replaces them with a smaller pattern or number that represents that pattern (McGeoch).

P	W	C_1	C_2
.40	not	110	11
.35	save	00	11111
.14	the	01	001
.06	trust	111	111
.05	queen	10	10

Figure 1. Two codes for the same set of source words. The first is a prefix code, the second is not.

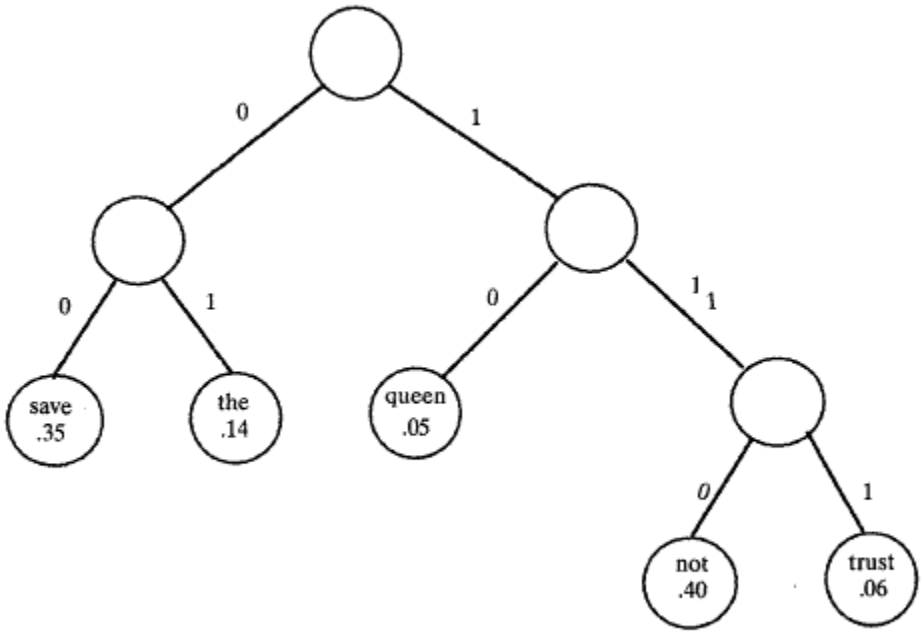

Figure 2. An encoding tree for code C_1.

Pictures taken from reference #1, Page 494-495

This is valuable when compressing large files with repetitious data, such as text files. The downside is that if the probability of each word or pattern is equal, then the efficiency drops dramatically (Lelewer and Hirschberg).

Lossy compression is utilized in various image compression. If certain colors are reduced in quality or removed altogether, the human eye cannot tell the difference between the compressed version and the original version.

VIII. Literature Cited

1. McGeoch, Catherine C. "Data Compression." The American Mathematical Monthly 100: 493-497. 31 Oct. 2008 <http://www.jstor.org/stable/2324310?seq=1&Search=yes&term=data&term=compression&list=hide&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3Ddata%2Bcompression%26x%3D1%26y%3D10%26wc%3Don&item=15&ttl=16184&returnArticleService=showArticle&resultsServiceName=doBasicResultsFromArticle>.

2. Data-Compression.com. EEF. 31 Oct. 2008 <http://www.data-compression.com/index.shtml>

3. Lelewer, Debra A, and Daniel S Hirschberg. "Data Compression." Data Compression. University of California. 31 Oct. 2008 <http://www.ics.uci.edu/~dan/pubs/DataCompression.html>. Research Paper