Data Compression through Duplicate Elimination and Tagging

Jeffrey Thomas, Computer Systems Lab Period 2

Abstract

Data compression is a valuable tool to save memory and time when sending data between computers. I believe to have created a new method to compress data based on simple probability and the concept of effective infinite data.

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. Effective methods of data compression are vital in the current computing world. In theory, the algorithm should be able to compress large data files to an optimal proportion any time it runs. 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.

Methodology

In order to deal with large groups of data, an ArrayList of ArrayLists of Strings are used. In this way, none of the ArrayLists will reach their maximum capacity. The main factor for success will be the amount of data compressed. The data files it uses will be randomly generated to simulate large files.

The algorithm works by finding patterns of bits where the first half is identical to the second half.

```
01100110
10001000
```

Fig 1. Duplicate bit patterns

```
00101100
11100111
```

Fig 2. Not Duplicate patterns

The second half is deleted, and a marker noting its former position is placed at the beginning of the stream of bits. The process is repeated for the rest of the stream.

Expected Results

With a sufficiently large file, ¼ of the bit patterns will be duplicates. Factoring in the length of the markers, an average of 12.5% of the original data will be compressed. This proportion should hold for any type of data, and for almost every run of the algorithm. With further manipulation, that percent could be increased, possibly asymptoticly.