University of Information Technology

Multimedia project

Compression algorithms

Truong Phuc Anh 14520040
Lam Han Vuong 14521106
Trieu Trang Vinh 14521097

January 7, 2018

Contents

List of Figures

List of Tables

1 Introduction

2 Dataset

3 Methods

3.1 Run-length coding

Basic idea

Examples

Pseudo-code

3.2 Shannon-Fano coding

Input

Set of symbol S.
The document need to compress.

Output

The compressed document.
The table of frequency (or number of times symbol appears in the document).

Basic idea

1. For a given list of symbols, calculate table of probabilities or frequency counts for the document

2. Sort the lists of symbols according to frequency (descending order)

3. Divide the list into two parts, with the total frequency counts of the left part being as close to the total of the right as possible.

4. The left part of the list is started with the code 0, and the right part is started with code 1.

5. Recursively apply the steps 3 and 4 to each of the two halves, subdividing groups and adding bits to the codes until each symbol has become a corresponding code leaf on the tree.

Example

The document need to compress: "ABBACAABCECAABADDDE".
Set of symbols S = A, B, C, D, E.
Encoded message: "00010100100000011011110000001001101101110111".
Decoded message: "ABBACAABCECAABADDDE".

Symbol	Count	Probability
A	7	0.37
B	4	0.21
C	3	0.16
D	3	0.16
E	2	0.11

Table 1: Calculate the table of frequency (descending oder)
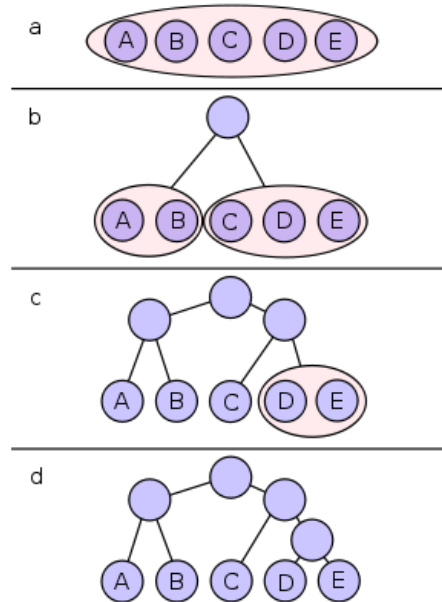
Figure 1: Divide the list of symbols and assign code

Symbol	Code
A	00
B	01
C	10
D	110
E	111

Table 2: The final code of symbols

Pseudo-code

You can find pseudo-code (Python style) for Shannon-Fano encoding and decoding in *pseudo-code/shannon-fano.py*

3.3 Lossless jpeg

Basic idea

Examples

Pseudo-code

4 Experiments

5 Conclusions

6 References

7 Appendices