THE SHANNON – FANO ALGORITHM

Student VICTORIA CUPEȚ
Student DAN CREȚU
Student ROLAND DRĂGOI
Student BOGDAN DIACONU
Faculty of Economic Cybernetics, Statistics and Informatics,
Academy of Economic Studies

1.Introducing Data Compression

Over the time, mankind has been trying to reduce, as possible, the effort made in its work. The information retrieval was not avoided, too.

The first step for the fast retrieval was made by using computers and, implicitly, it was necessary to turn information into digital format. To find it more easily, information must be stocked, eventually, transmitted through a communication channel.

Stocking and transmission have some costs. More information processed, higher the costs are. In spite of this reason, the stocking is not in the most compact form. There are used methods which make information more easily to access, but they use more space to represent it. Thus appeared as necessary to reduce the space occupied by information. This is known as data compression.

The notion of data compression is referring to miscellaneous algorithms and programs which solve this problem of space. A compression algorithm is used to turn data into a more compact format from, so called, easy-to-use format. Decompression is the method which transform data into their original format.

In this project is studied the compression-decompression Shannon-Fano algorithm.

2. Brief History

The ‘40s are considered as the pioneer’s work years for *information theory* domain. Notions like redundancy, entropy and information content have been studied with more interest. The founder of this new science is Claude E. Shannon himself.

Master in mathematics at MIT in 1940, *Claude E. Shannon* had been pondering over elements of information theory since 1930. These ideas have interested him seriously from academy year 1940-1941, which had found him at Research Fellowship, part of Advanced Studies Institute from Princeton. The result of his researches had been published in “A Mathematical Theory of Communication", which had appeared in 1948.[Shan49]. This book proves that all information sources – telegraph, human speaking, television and many more – have an associate source rate which is measured in bytes per second. On the basis of his observation, Shannon had built a model: the source-encoder-channel-decoder-destination model which is fundamental for all our days communications.(!work.htm) Claude Shannon introduced a revolutionary, proceeded from probabilities way to think communication, creating a true mathematic theory for many information notions. His ideas were assimilated rapidly and developed in two directions: information theory and codes theory.

Master professor at Massachusetts Institute of Technology, Robert M. Fano has pioneer work contribution in the area of time-sharing computers systems. His work to information theory has been rewarded with the IT Society’s Claude E. Shannon Prize in 1976.

The Shannon-Fano algorithm has been developed independently by Claude E. Shannon şi Robert M. Fano in two different books, which have appeared in the same year, 1949.

3. The Algorithm Description

Claude Shannon has introduced some mathematic concepts while he was looking for solutions for communication base problems.

The most important concept is the *entropy*. Assumed from thermodynamics, entropy signifies the quantity of information reported to an element of the message. In "A Mathematical Theory of Communication", Shannon defined this notion as it follows: if a source produce symbols with probabilities p_1, p_2, p_3,…,p_n, then his entropy is given by:

$$H = -\sum_{i=0}^{n} p_i \log_2 p_i$$

He also demonstrated that the best rate of compression is at least equal with the source entropy.[Shan48]

The Shannon Fano algorithm does not produce the best compression method, but is a pretty efficient one. Using for compression the occurrence probabilities of each symbol, it is a part of the statistical algorithms class. Huffman an Lempel-Ziv are part of this class too. This is a „loss-less" algorithm, that is the original codified information is retrieved entirely in decodified file (with minimum losses). This property is used to compress data in which every bit matters (text processing, spreadsheets, databases).

Claude Shannon and Robert Fano have established three rules which allow the compression of a message:

- every code has a different number of bits;
- the corresponding code of a symbol which has a lower occurrence probability within the message would be stored by a larger number of bits; the code of a symbol with a higher frequency will be stored by fewer bits.
- the codes of different length are not a barrier for the decompression algorithm's identifying the initial symbols.

The steps of the algorithm are :

S1. The message symbols are sorted in descending order by frequency and moved into a table.

S2. The table is split in two parts so that the sum of frequencies of each one be as closest as possible.

S3. 0 is attached to the first table part and 1 is attached to the second one.

S4. Steps 2 and 3 repeat until we obtain groups of singular elements.

We notice that the algorithm involves scanning the message twice: first for obtaining the frequencies and second for building the start table.

4. Compression

The principle consists in re-codifying data like that from standard ASCII files on less than 8 bits. The characters that appear very rarely in the text will be codified using more bits than for those ones that appear frequently.

The codification will take place in three main phases:

The **first phase** consists in building a table with the frequencies of occurrence of all the characters from the source file and for that it is necessary to read the file entirely. The data structure used in the program for implementing the table will be an array of structures. For every ASCII character from the file there will be a corresponding article in the table ant this is the reason why the array's dimension is 255. The fields of an article of the table are:

- the frequency of occurrence of that character in the source file;
- the code found for the character;
- the ASCII code of the character;

- a boolean variable which indicates whether or not that character has been codified.

Before beginning calculating the frequencies of occurrence, the table is initialized, that is all the articles are attributed the values *0*, "", *ASCII_code* and *False* respectively.

After the frequencies are calculated and written in the table, the table of frequencies is obtained, which is sorted in ascending order by the ASCII code. This table must be sorted in descending order by the frequencies of occurrence. This is necessary because the characters codification algorithm generates the codes on sizes that are reverse proportional to the frequencies of occurrence. The next table is obtained:

Character	ASCII code	Frequency
...		
LF	10	6
...		
CR	13	6
...		
1	49	17
2	50	33
3	51	20
4	52	29
...		
A	97	33
...		
D	100	48
...		
F	102	47
...		
S	115	49
..		

The Frequencies table
(sorted in ascending order by ASCII code)

Character	ASCII code	Frequency
s	115	49
d	100	48
f	102	47
2	50	33
a	97	33
4	52	29
3	51	20
1	49	17
LF	10	6
CR	13	6
...	...	0

The Frequencies table
(sorted in descending order by frequencies)

In the **second phase**, all the characters are uniquely codified, using an elaborated algorithm. The procedure for determining the codes will emphasize a recursive algorithmic structure.

The current ensemble is to be divided in two sub-ensembles whose cumulated frequencies are almost equal. A 0 will be added to the code of the upper sub-ensemble and a 1 to the code of the lower one.

The algorithm repeats for each of the remaining ensembles, the stopping condition being that the current ensemble to consist in one character.

In order to separate the current ensemble, we shall calculate the theoretic half of the cumulated frequencies; for each one of the variables that are being used (*CurrentSum* and*PreviousSum*) we shall search for the biggest distance to the theoretic half.

<table>
<tr><td rowspan="3">s 144
d A_1
f '0'</td><td colspan="4">s 49 '00' A_3</td></tr>
<tr><td rowspan="2">95 '01' A_4</td><td colspan="3">d 48 '010' A_7</td></tr>
<tr><td colspan="3">f 47 '011' A_8</td></tr>
<tr><td rowspan="6">2
a 144
4
3 '1'
1
LF A_2
CR</td><td rowspan="2">66 '10' A_5</td><td colspan="3">2 33 '100' A_9</td></tr>
<tr><td colspan="3">a 33 '101' A_{10}</td></tr>
<tr><td rowspan="4">78 '11' A_6</td><td>49 '110' A_{11}</td><td colspan="2">4 29 '1100' A_{13}</td></tr>
<tr><td></td><td colspan="2">3 20 '1101' A_{14}</td></tr>
<tr><td rowspan="2">29 '111' A_{12}</td><td colspan="2">1 17 '1110' A_{15}</td></tr>
<tr><td>12 '1111' A_{16}</td><td>LF 6 '11110' A_{17}</td></tr>
<tr><td colspan="4"></td><td>CR 6 '11111' A_{18}</td></tr>
</table>

The coding process with Shannon- Fano algorithm

The resulting codes will be written in the table, and the field that shows the status of encoding will be updated in proper manner, resulting:

Character	ASCII Code	Codified/ Uncodified	Frequency	Code
s	115	true	49	00
d	100	true	48	010
f	102	true	47	011
2	50	true	33	100
a	97	true	33	101
4	52	true	29	1100
3	51	true	20	1101
1	49	true	17	1110
LF	10	true	6	11110
CR	13	true	6	11111
...	...	false	0	-

The table with the associated codes for each character (sorted by frequency)

Finally, in the **third phase**, these codes are being used in order to "translate" the initial information and to build an ensemble of compressed data. Because it is necessary that the table should be sorted by ASCII codes to guarantee an easy access at articles, the useful information from now on (the code and the coding status) will be copied into another table which, being already initialized, is sorted by ASCII codes.

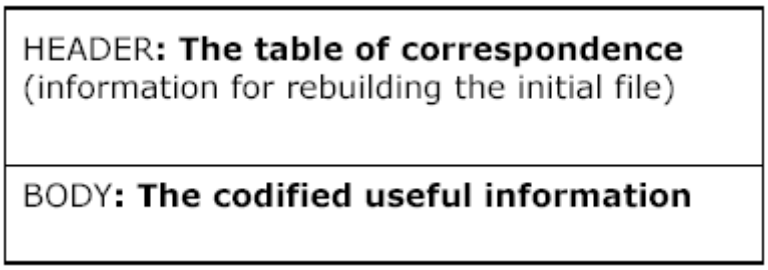
The next step will be creating the table of correspondence and writing it in the compressed file. The table of correspondence shows like a thread of bytes of length equal to the number of different characters of the source file multiplied by the length of the correspondence sequence.

The correspondence sequence takes at most 4 bytes:

- first byte contains the ASCII code of the character that will be converted;
- 6 bits of the second byte contain the number of bits used for coding (maximum 16 bits);
- 2 (8-6) bits of the second byte + the third byte + the fourth byte eventually will contain the character code.

After writing the table of correspondence in the compressed file, the source file will be scanned for the second time, in order to codify the information contained, using the code obtained in the previous step.

Finally, the compressed file will consist of two parts:



5.Decompression

The phases of decompression are:

D1. creating the decodification tree
D2. the actual decodification using the decodification tree

D1. Creating the decodification tree
1) The root is created;
2) Every bit of each character's code will be added to the tree;

D2. The steps of the actual codification
1) The file is read bit by bit;

2) Root is used as starting point;

3) If the bit is : - 1 - there will be a right shifting, if there is a node ;

-the next bit is read;

- 0 - there will be a left shifting, if there is a node ;

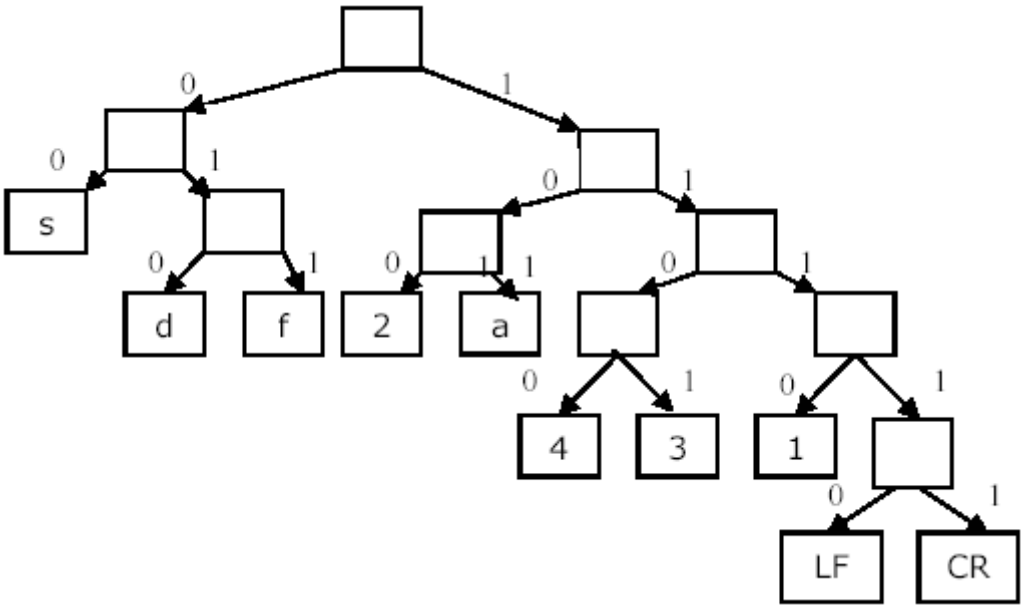
-the next bit is read;

4)If there isn't any node – the character is written as a leaf;

- the next bit is read;

- we start back from the root;

The table's associated tree is:



For example, the next sequence:

011 100 00 1101 11111 11110

Is interpreted as:

f 2 s 3 CR LF

To store this sequence uncompressed we need 6x8=48 bits
The same sequence, but compressed will take 22 bits.

6. Conclusion

Shannon -Fano algorithm is one of the first algorithms and perhaps because of that it's performance is not so good.

Although, this algorithm represents the starting point for a wider class of algorithms, that have the same working principle(statistical algorithms).

The principal concept – high frequency character have a short code and low frequency characters have a long code – of Shannon -Fano algorithm is the starting point of many algorithms.

Compression is a domain that involves a lot of work and imagination, in order to improve our activities' performances.

References

[Shan48] Shannon, Claude *A Mathematical Theory of Communication*, The Bell System Technical Journal, nr. 379-636. Bell Laboratories, 1948

[Hoff97] Hoffman, Roy *Data Compression in Digital Systems,* Chapman&Hall,1997

[Ivan98] Ivan, Ion
Verniş, Daniel *Compresia de date,* Ed. Cison, Bucharest, 1998

[Nels92] Nelson, Mark *La compression de données. Texte. Images. Sons*, Ed. Dunod, Paris,1992

http://www.lwithers.demon.co.uk/u/notes/2-cy-d8/2001-05-03-1000.html

Information Theory. Review of Part 1; Source/Channel Matching

http://www.faqs.org/faqs/compression-faq/part2/section-1.html

[70] Introduction to Data Compression (long)

http://www.alphabeta-net.com/en/Fano_Shannon.html

Méthode de Fano-Shannon

http://www.nae.edu/nae/naepub.nsf/Members

Membership Directory

http://www.eee.bham.ac.uk/WoolleySI/All7/intro_2.htm

The Need For Compression

http://cm.bell-labs.com/cm/ms/what/shannonday/work.html

The Significance of Shannon's Work

http://golay.uvic.ca/awards/shannon.html

Claude E. Shannon Award

http://www.research.att.com/~njas/doc/shannonbio.html

Biography of Claude E. Shannon

*All the Internet addresses from above were checked on November 30th 2001.