

## Contents

Introduction .....	1
Data compression .....	2
Rationale .....	2
Advantages .....	2
Shannon Fano Coding .....	3
Method .....	3
Principle .....	3
For example .....	3
Exercise .....	5
Conclusion .....	8
Reflection .....	8
Reference .....	8

## Introduction

This chapter introduces the concept of data compression, a discrete memoryless source, the Shannon Fano Coding,  $H(X)$  and the average codeword length.

Data compression is one of the enabling technologies for each aspect of the multimedia revolution. Compression is used just about everywhere. There are the two major types of compression algorithms: lossless compression and lossy compression. Lossless compression is used for applications that require an exact reconstruction of the original data, it is typically used for text. While lossy compression is used when the user can tolerate some differences between the original and reconstructed representations of the data, it is used for images and sound where a little bit of loss in resolution is often undetectable, or at least acceptable.

Discrete source of information is uniquely defined by its probabilistic model two classes of sources can be distinguished: Memoryless source and Markov source. Entropy  $H(X)$  is a measure of information generated by a memoryless source  $X$ .

Shannon–Fano coding, named after Claude Elwood Shannon and Robert Fano, is a technique for constructing a prefix code based on a set of symbols and their probabilities.

## Data compression

### Rationale

Data compression is the process of encoding, restructuring or modifying data in order to reduce its size. It involves re-encoding information using fewer bits than the original representation. An important element in the design of the data communications algorithms is the modeling of the data. We distinguish between lossless algorithms and lossy algorithms.

Lossless algorithms can reconstruct the original message exactly from the compressed message. Lossless algorithms are typically used for text. It is important that the reconstruction is identical to the original text, as very small differences can result in statements with very different meaning. Therefore, it is not advisable to allow any differences to appear in the data compression process.

Lossy algorithms can reconstruct an approximation of the original message, in other words user can tolerate some difference between the original and reconstructed representations of the data. Lossy algorithms are typically used for images and sound where a little bit of loss in resolution is often undetectable, or at least acceptable. Lossy is used in an abstract sense, however, and does not mean random lost pixels, but instead means loss of a quantity such as a frequency component, or perhaps loss of noise. For example, one might think that lossy text compression would be unacceptable because they are imagining missing or switched characters. Consider instead a system that reworded sentences into a more standard form, or replaced words with synonyms so that the file can be better compressed. Technically the compression would be lossy since the text has changed, but the “meaning” and clarity of the message might be fully maintained, or even improved.

### Advantages

The entropy of a discrete memoryless source, which is the average value of self-information, can be interpreted as the amount of uncertainty associated with the source or, in other words, the entropy of the source is the amount of information that needs to be provided to remove the uncertainty in the source.

Advantages of Data Compression:

Reduced Storage Space, Faster Data Transfer, Bandwidth Efficiency (conserves bandwidth, reduce cost), Resource Optimization(memory-constrained), Cost Savings, Security and Privacy.

Advantages of a Discrete Memoryless Source

Information Theory (designing efficient communication systems and encoding schemes) ,  
Mathematical Simplicity(simpler to model and analyze mathematically), Foundation for  
Coding Theory(efficient encoding and decoding techniques)

In summary, data compression is primarily focused on practical applications, such as reducing storage and transmission costs, optimizing resource usage, and enhancing the user experience. On the other hand, a discrete memoryless source is a theoretical concept used in information theory to model the behavior of data sources and serves as the foundation for

efficient communication system design. While data compression and discrete memoryless sources serve different purposes, they are interrelated in that compression techniques are used to reduce the size of data generated by such sources for practical applications.

## Shannon Fano Coding

### Method

A Shannon–Fano tree is built according to a specification designed to define an effective code table. The actual algorithm is simple:

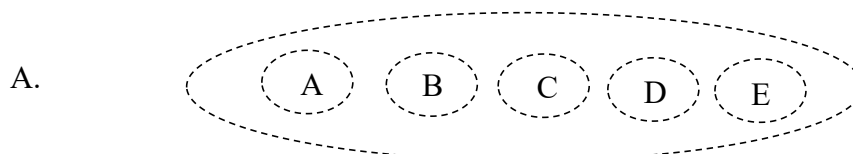
1. For a given list of symbols, develop a corresponding list of probabilities or frequency counts so that each symbol's relative frequency of occurrence is known.
2. Sort the lists of symbols according to frequency, with the most frequently occurring symbols at the group1 and the least common at the group2.
3. Divide the list into two parts, with the total frequency counts of the group1 part being as close to the total of the group2 as possible.
4. The group1 part of the list is assigned the binary digit 0, and the group2 part is assigned the digit 1. This means that the codes for the symbols in the first part will all start with 0, and the codes in the second part will all start with 1.
5. Repeat 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.

### Principle

Shannon–Fano coding, is a technique for constructing a prefix code based on a set of symbols and their probabilities. It does guarantee that all codeword lengths are within one bit of their theoretical ideal  $I(x) = -\log P(x)$ .

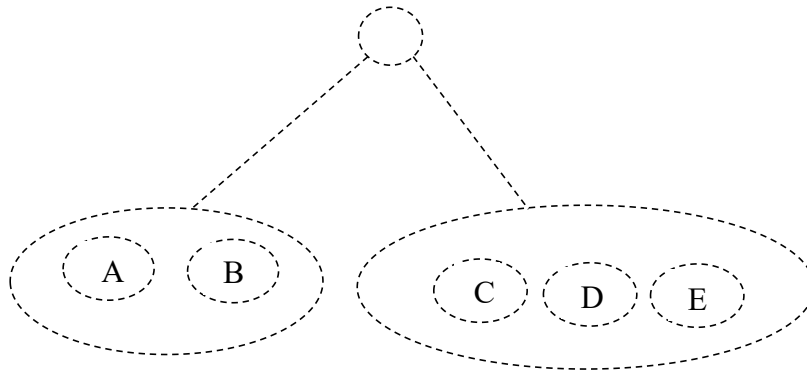
In Shannon–Fano coding, the symbols are arranged in order from most probable to least probable, and then divided into two sets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned; symbols in the first set receive "0" and symbols in the second set receive "1". As long as any sets with more than one member remain, the same process is repeated on those sets, to determine successive digits of their codes. When a set has been reduced to one symbol, of course, this means the symbol's code is complete and will not form the prefix of any other symbol's code. The algorithm works, and it produces fairly efficient variable-length encodings; when the two smaller sets produced by a partitioning are in fact of equal probability, the one bit of information used to distinguish them is used most efficiently.

### For example

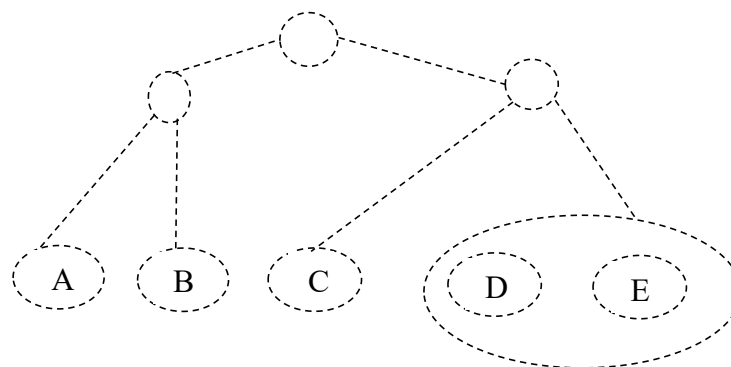


*Figure 1 Step 1 - Shannon Fano Coding Method*

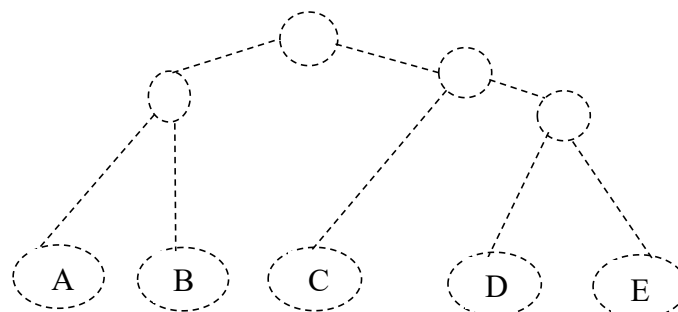
B.

*Figure 2 Step 2 - Shannon Fano Coding Method*

C.

*Figure 3 Step 3 - Shannon Fano Coding Method*

D.

*Figure 4 Step 4 - Shannon Fano Coding Method*

**Exercise**

Symbol	A	B	C	D	E	F	G	H
Prob	0.15	0.5	0.15	0.02	0.08	0.01	0.01	0.08

*Chart 1 - Shannon Fano Coding Exercise*

Sort the Inputs Via Highest Probability Symbol on the Left

Symbol	B	A	C	E	H	D	F	G
Prob	0.5	0.15	0.15	0.08	0.08	0.02	0.01	0.01

*Chart 2 - Shannon Fano Coding Order*

Split the Symbol Alphabet as close to balance the probabilities as closely as possible on the Group1 and Group2.

Group1

Symbol	B
Prob	0.5

*Chart 3 - Shannon Fano Coding Group 1*

Group2

Symbol	A	C	E	H	D	F	G
Prob	0.15	0.15	0.08	0.08	0.02	0.01	0.01

*Chart 4- Shannon Fano Coding Group 2*


Assign a 0 to the Group1 and a 1 to the Group2

Group1

	0
↓	
Symbol	B
Prob	0.5

*Chart 5- Shannon Fano Coding Assign a 0 to the Group1*

Group2



Symbol	A	C	E	H	D	F	G
Prob	0.15	0.15	0.08	0.08	0.02	0.01	0.01

*Chart 6- Shannon Fano Coding Assign a 1 to the Group2*

Repeat the process for all branches

Symbol	Prob	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7
B	0.5	0	0	/	/	/	/	0
A	0.15	0	1	0	/	/	/	01
C	0.15	1	0	0	/	/	/	100
E	0.08	1	0	1	/	/	/	101
H	0.08	1	1	0	/	/	/	110
D	0.02	1	1	1	0	0	/	11100
F	0.01	1	1	1	0	1	/	11101
G	0.01	1	1	1	1	0	/	11110

*Chart 7 - Shannon Fano Coding Repeat*

Symbol	Prob	Shannon Fano Code
B	0.5	0
A	0.15	01
C	0.15	100
E	0.08	101
H	0.08	110
D	0.02	11100
F	0.01	11101
G	0.01	11110

*Chart 8- Shannon Fano Coding Result*

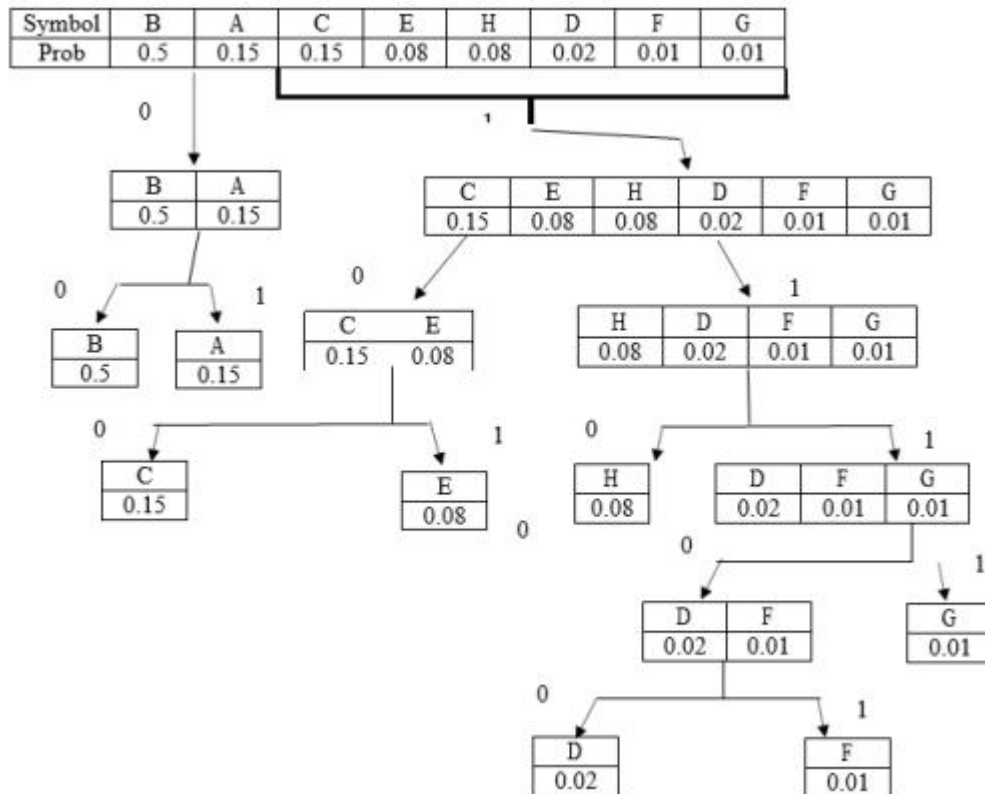


Figure 9 - Shannon Fano Coding Repeat

$$H(X) = - \sum_{i=1}^n (P_{xi}) * \log_2 ((P_{xi}))$$

Using the provided probabilities:

$$H(X) = -[0.5 * \log_2 (0.5) + 0.15 * \log_2 (0.15) + 0.15 * \log_2 (0.15) + 0.08 * \log_2 (0.08) + 0.08 * \log_2 (0.08) + 0.02 * \log_2 (0.02) + 0.01 * \log_2 (0.01) + 0.01 * \log_2 (0.01)]$$

$$H(X) = 2.4764 \text{ bits/symbol}$$

$$L_{avg} = \sum_{i=1}^n (P_{xi}) * \text{length of code for } X_i.$$

$$L_{avg} = 0.5*1 + 0.15*2 + 0.15*3 + 0.08*3 + 0.08*3 + 0.02*5 + 0.01*5 + 0.01*5 = 1.93 \text{ bits/symbol.}$$

So, for the Shannon Fano coding scheme applied to the given alphabet, the entropy  $H(X)$  is approximately 2.4764 bits/symbol, and the average codeword length ( $L_{avg}$ ) is approximately 1.69 bits/symbol. These values provide insights into the efficiency of the coding scheme in terms of representing the source symbols.



## Conclusion

### Reflection

This article main include: the concept of data compression, the advantage of data compression, the concept of a discrete memoryless source, the concept of Shannon Fano Coding, the method of Shannon Fano Coding,  $H(X)$  and the average codeword length.

### Reference

*Data Compression introduction [online]*

Available at : <https://www.cs.cmu.edu/~guyb/realworld/compression.pdf>

Available at : [https://books.google.ie/books?hl=zh-CN&lr=&id=3DFHDgAAQBAJ&oi=fnd&pg=PP1&dq=data+compression&ots=gGQmgBPg03&sig=whnxwCzF9r8u4LjvKGdkXH0RPns&redir\\_esc=y#v=onepage&q=data%20compression&f=false](https://books.google.ie/books?hl=zh-CN&lr=&id=3DFHDgAAQBAJ&oi=fnd&pg=PP1&dq=data+compression&ots=gGQmgBPg03&sig=whnxwCzF9r8u4LjvKGdkXH0RPns&redir_esc=y#v=onepage&q=data%20compression&f=false)

*Discrete Memoryless Source [Online]*

Available

at : [https://www.google.com/search?q=a+discrete+memoryless+source&oq=a+discrete+memoryless+source&gs\\_lcrp=EgZjaHJvbWUyBggAEEUYOTIJCAEQABgTGIAE0gEIMTI0OWowajeoAgCwAgA&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:7746d086,vid:xEj7ToJ2BSI,st:0](https://www.google.com/search?q=a+discrete+memoryless+source&oq=a+discrete+memoryless+source&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIJCAEQABgTGIAE0gEIMTI0OWowajeoAgCwAgA&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:7746d086,vid:xEj7ToJ2BSI,st:0)

*Average Code Length | Data Compression [Online]*

Available at : <https://www.youtube.com/watch?v=DlBBYKGY9Gw>

*Shannon – Fano Code [Online]*

Available at : [https://ecehithaldia.in/teaching\\_material/Shanon-Fano1586521731.pdf](https://ecehithaldia.in/teaching_material/Shanon-Fano1586521731.pdf)

*Shannon – Fano Coding [Online]*

Available at : <http://www.faadooengineers.com/online-study/post/eee/principle-of-communication/1281/shannon-fano-coding>

*Shannon – Fano Coding [Online]*

Available at : <https://youtu.be/dJCcklOgsIA?si=qRFItSqzk5RiB-p>