CHAPTER 43

Academic research for implementation of python language programs for information measurement and coding for data compression

10.56238/pacfdnsv1-043

Paulo Cesar de Souza Cavalcante
Professor at the School of Technology (EST) of the Amazonas State University (UEA)
Av. Darcy Vargas, 1200 – Parque 10 de Novembro - Manaus – AM – Brasil - CEP 69050-020
Title: Master in Electrical Engineering in Telecommunications from the Federal University of Pernambuco – UFPE
E-mail: pcscavalcante@gmail.com

Williams Cavalcante de Oliveira
Student at the School of Technology (EST) of the State University of Amazonas (UEA)
Av. Darcy Vargas, 1200 – Parque 10 de Novembro - Manaus – AM – Brasil - CEP 69050-020
Title: Electronic Engineering Graduating Student
E-mail: wco.ele18@uea.edu.br

ABSTRACT
This article presents the expanded abstract containing information about the academic research for implementation of programs in Python language for information measures and coding for data compression. The research was conducted, at the Amazonas State University, by the student of Electronic Engineering course Williams Cavalcante Oliveira, under the guidance of Professor Paulo Cesar de Souza Cavalcante, and had its abstract presented in the 1st Amazon Stem Academy Conference 2021. The content of the presented abstract, transcribed in this paper, exposes brief theoretical review of information measurement calculation methods and methods for statistical and dictionary-based data compression encoding. Besides other information concerning the research, it also presents, as its final product, the Python language modules developed for the execution of the algorithms of the methods studied, for performing the calculations and encodings that were the object of the research.

Keywords: Information Measurement, Data compression encoding, Python Language.

1 INTRODUCTION

The present research aimed to develop modules in Python language for performing information measurement calculations and for data compression, addressed in part of the content of the subject Theory of Information and Coding (TIC). Huffman h To this end, a literature review was conducted on TIC course material and material for basic learning of the Python programming language. Program modules were developed for calculating information measures and modules for implementing compression algorithms, based on statistics: Shannon-Fano, Huffman, Arithmetic, and based on dictionaries: Lempel Ziv. The developed modules will allow TIC students, or others interested, to verify in practice their theoretical realizations of the functions needed for digital communication over noisy channels or for their storage in compressed form.

2 OBJECTIVES

The general objective of the project is to create and make available additional tools for consolidation of knowledge during the course of the Theory of Information discipline. The specific objectives, which

contribute to the achievement of the general objective, are: teamwork exercise; learning the Python programming language; consolidation of the didactic knowledge transmitted about information measures and data compression.

3 MATERIALS AND METHODS

The project was developed in compliance with the following stages: 1) indication by the Advisor of basic bibliography for research; 2) accomplishment of research by the student in physical documentation and material available on the Internet; 3) elaboration by the student of a brief theoretical reference on Information Measurement and data compression using the following codifications: Shannon-Fano, Huffman, Arithmetic and Lempel-Ziv; 4) Implementation by the Oriented Student of modules in Python language to perform Information Measurement calculations and encoding using the specified encodings. 5) Evaluation and guidance by the Advisor in each of the topics specified in the third step, as they were completed; 6) Joint preparation by the Advisor and the student of this Expanded Summary, and 7) Preparation by the student, with the Advisor's supervision, of the Project Presentation at a conference.

The bibliography indicated by the Advisor in the first stage was: Information Theory Class Notes [Cavalcante 2021]; video of the minicourse Mathematics with Python [INPE 2020]; site for access to Python Documentation [Python Software Foundation 2021].

3.1. THEORETICAL BASIS

In this subsection the theoretical basis on Information Measures and Coding for Compression of Digital Data will be exposed, i.e. data that is expressed using the binary numbering base, where there are only two symbols: 0 and 1. The following material was extracted from the Information Theory Class Notes [Cavalcante 2021].

I) Information Measure

Information is everything that is produced by a source to be transferred to the user. Regarding the measure of information one can consider two points of view: from the User, the measure of information is related to uncertainty (regarding the message that was transmitted), and from the Source, the measure of information is an indication of the freedom of choice exercised by the source when selecting a message. If the source has many different messages: the probability of each message tends to decrease with the number of messages. The user will have more doubts (more uncertainty, more information) regarding the message that will be chosen. For example, if the source has only one possible message: The probability will be maximum, equal to 1. The user will have no doubt (no uncertainty, no information) in relation to the message that will be chosen.

A) *Hartley's Information Measure or Self-Information*

Hartley (1928) proposed as a measure of the amount of information provided by the observation of a discrete random variable X. The information of a single symbol, I(X), is suggested by logb K, where K is the number of possible values of X. The probability of occurrence of one of the possible values of X is PX=1/K. Thus, K= 1/PX, i.e. K is the inverse of the probability of occurrence of a symbol. Hartley's information measure:

$$I(X) = log_b\ K = log_b \left(\frac{1}{P_X} \right) = - log_b\ P_X \qquad (1)$$

Thus Hartley's information measure can be defined as a logarithmic quantity linked to the inverse of the probability of an event. The base of the logarithm used, defines the unit of the information measure. If base 2 is used, as in digital communications, the unit will be bit. To exemplify, let the probabilities of emission by a source of bits 0 and 1, respectively: P0 = 1/4; P1 = 3/4, we have: Information transported by the digit 0: I0 = - log2 1/4 = 2 bits and Information transported by the digit 1: I1 = - log2 3/4 = 0.41 bits.

B) *Shannon Information Measure or Entropy*

Shannon (1948) defined that in general, if the i-th value of X has probability PX (xi), then the Hartley information log 1/PX(xi) = - log PX(xi) for this value should be pondered by PX(xi), giving:

$$H(X) = \sum_i P_X(x_i) \log_2 \frac{1}{P_X(x_i)} = -\sum_i P_X(x_i) \log_2 P_X(x_i) \qquad (2)$$

Shannon's measure could be considered Hartley's average information. Shannon called this measure of information entropy. The entropy of a source means that on average we expect to get H bits of information per symbol. Rewriting the entropy formula as:

$$\mathbf{H(P_1, P_2, ..., P_M) = \sum_{j=1}^{M} P_j \log_2 \frac{1}{P_j} = - \sum_{j=1}^{M} P_j \log_2 P_j} \qquad (3)$$

To exemplify, suppose a source X emits four symbols x0, x1, x2, and x3 with probabilities 1/2, 1/4, 1/8, and 1/8, respectively. The uncertainty, or entropy H(X) is given by: H(X) = (1/2) log 2 + (1/4) log 4 + (1/8) log 8 + (1/8) log 8 = 1.75 bits/symbol.

II) Encoding for Data Compression

The data compression process is performed by algorithms that receive M messages, sequences of bits of length N, and encode them for transmission or storage in M messages, whose length is less than N bits of the original messages, without loss of information (Lossless compression techniques). Statistical, or entropy-based, models need to know the statistics of occurrence of the symbols to be coded. Adaptive, or

dictionary-based, models perform the compression without needing the statistics of the source. The techniques covered in this research are shown in Chart 1 below.

Chart 1. Lossless Compression Techniques

Examples of Statistical Models	Examples of Dictionary-Based Techniques
Codes: Shannon-Fano, Huffman and Arithmetic	Lempel-Ziv(LZ) Codes: Lempel-Ziv (LZ 77), Lempel-Ziv (LZ 78) and Lempel-Ziv-Welch (LZW)

The average length of code words resulting from encoding for data compression is determined by:

$$\overline{L} = \sum_{i=0}^{M-1} p_i l_i \qquad (4)$$

where l_i is the number of bits of the codeword corresponding to the symbol i, which occurs with probability p_i. Coding efficiency can be defined by the ratio between the entropy of the source or message and the average length of the code words:

$$\eta = \frac{H(S)}{\overline{L}} \qquad (5)$$

The compression ratio Tc of a compressor code is defined by the expression below:

$$T_c = \frac{\text{Qde bits texto sem compressão - Qde bits texto com compressão}}{\text{Qde bits texto sem compressão}} \text{ x } 100 \qquad (6)$$

TRADUÇÃO:
Qde bits texto sem compressão – Qde bits texto com compressão: Bit rate uncompressed text – bit rate compressed text
Qde bits texto sem compressão: Bit rate uncompressed text

3.2. MODULE DEVELOPMENT IN PYTHON

The development of the modules in Python language was carried out in the platform *Intergrated Development Environment* (IDE) PyCharm, in production on a computing platform with Windows 10 operating system. The following Python modules were implemented for entropy and compression calculation using encoding by the respective algorithms: Entropy; Shannon-Fano; Huffman; Arithmetic; Lempel-Ziv LZ 78. The modules developed can be found in the document Python Module Listings [Oliveira and Cavalcante 2021a].

4 RESULTS

Message compression tests were performed by the developed Python modules. The following Table 2 and Table 3 present a consolidation of the data extracted from the execution of the modules, contained in the document Python modules test results [Oliveira and Cavalcante 2021b]. The data exposed is the actual coding and measurement parameters exposed in the theoretical foundation of this work.

Table 2. Ascii message compression test (8 bits/symbol): demonstration for first amazon stem academy conference

Simbolos texto			Código Shannon Fano			CódigoHuffman			Código LZ78		
s	qde s	p(si)	Pal.-codigo	bits/s	tot bits/s	Pal.-codigo	bits/s	tot bits/s	Segmentos M	Pal.-codigo	bits/s
A	7	7/54	000	3	21	100	3	21	'DE'	00001100100	11
ESP	6	1/9	001	3	18	101	3	18	'MO'	00011101001	11
E	6	1/9	010	3	18	110	3	18	'NS'	00100001100	11
M	4	2/27	0110	4	16	0010	4	16	'TR'	00110101011	11
O	4	2/27	0111	4	16	0011	4	16	'AÇ'	00000110001	11
N	4	2/27	100	3	12	0100	4	16	'ÃO'	01000001001	11
S	4	2/27	1010	4	16	0101	4	16	' P'	00000001010	11
R	4	2/27	1011	4	16	111	3	12	'AR'	00000101011	11
T	3	1/18	1100	4	12	0111	4	12	'A '	00000100000	11
D	2	1/27	11010	5	10	00010	5	10	'FI'	00010100110	11
F	2	1/27	11011	5	10	01100	5	10	'RS'	00101101100	11
C	2	1/27	11100	5	10	01101	5	10	'T '	00110100000	11
Ç	1	1/54	111010	6	6	000000	6	6	'AM'	00000100111	11
Ã	1	1/54	111011	6	6	000001	6	6	'AZ'	00000101111	11
P	1	1/54	111100	6	6	000010	6	6	'ON'	00100101000	11
I	1	1/54	111101	6	6	000011	6	6	' S'	00000001100	11
Z	1	1/54	111110	6	6	000110	6	6	'TE'	00110100100	11
Y	1	1/54	111111	6	6	000111	6	6	'M '	00011100000	11
18	54	1,00	Tot. bits codificação ->		211	Tot. bits codificação ->		211	'AC'	00000100010	11
									'AD'	00000100011	11

Texto sem codificação
Total de bits
54 x 8 bits/s= 432 bits
Entropia do texto
(Fórmula 3)
H(P) = 3,87 bis/s

Código	Comp. Médio (Form. 4)	Eficiência Código (Form.5)	Taxa Compressão (Form. 6)
Shannon	3,9 b/s	99,21%	51,16%
Huffman	3,9 b/s	99,21%	51,16%
LZ78	11 b/s	35,18%	31,25%

Nota: O código LZ-78 não se utiliza da estatística da fonte, é baseado em dicionário (adaptativo)

'EM'	00010000111	11
'Y '	00111000000	11
'CO'	00001001001	11
'NF'	00100000101	11
'ER'	00010001011	11
'EN'	00010001000	11
'CE'	00001000100	11
	Tot. bits codificação ->	297

TRADUÇÃO TABELA2:
Símbolos texto: Text symbols
Código Shannon Fano: Shannon Fano Code
Código Huffman: Huffman Code
Código LZ78: LZ78 Code
Qdes: qtts
Pal. – código: Code Pal.
Segmentos M: M Segments
Texto sem codificação: Unencoded text
Total de bits: bitrate
Entropia do texto (Fórmula 3): Text entropy (Formula 3)
Código: Code
Comp. Médio: Medium Comp.
Eficiência Código: Code efficiency
Taxa compressão: Compression ratio
Nota: O código LZ-78 não se utiliza da estatística da fonte, é baseado em dicionário (adaptativo): Note: The LZ-78 code does not use source statistics, it is dictionary-based (adaptive)

Although there were no errors in the compression tests performed according to Tables 2 and 3, in other tests some non-conformities were verified in the Entropy (probability totalization), Arithmetic (decimal to binary conversion error) and Lempel-Ziv LZ-78 (suppression of the last message segment coding) modules, which are already under analysis for elimination.

Table 3. ASCII Message Compression Test (8 bits/symbol): asadacasa

Simbolos texto			Código Aritmético		Código LZ78		
s	qde s	p(si)	Intervalo Codificado	Pal.-codigo	Segmentos M	Pal.-codigo	bits/s
a	5	5/9	[0.365896, 0.365928)	**101110111**	´as'	**000011**	6
c	1	1/9	Conv.Pal.-cód em Dec	Dentro	´ad`	**000010**	6
d	1	1/9	0,101110111 (2)=	Intervalo?	´ac`	**000001**	6
s	2	2/9	0.365910(10)	**Sim**	´asa`	**010000**	6
4	**9**	**1,00**	**Tot. bits codificação ->**	**10**		**Tot. bits codificação ->**	**24**

Texto sem codificação
Total de bits
9 x 8 bits/s= 72 bits
Entropia do texto
(Fórmula 3)
H(P) = 1,65 bis/s

Código	**Comp. Médio (Form. 4)**	**Eficiência Código (Form.5)**	**Taxa Compressão (Form. 6)**
Aritmético	**10 b/s**	**-**	**86,11%**
LZ78	**6 b/s**	**-**	**51,16%**

Nota: O código LZ-78 não se utiliza da estatística da fonte, é baseado em dicionário (adaptativo)

TRADUÇÃO – TABELA 3
Símbolos Texto: Text Simbols
Código Aritmético: Arithmetic code
Pal. – Código: Pal. – Code
Segmentos M – M segments
Pal. – Código: Pal. – Code
Texto sem codificação: Unencoded text
Total de bits: bitrate
Entropia do texto: Text entropy (Formula 3)
Código: Code
Comp. Médio: Medium Comp.
Eficiência Código: Code efficiency
Taxa compressão: Compression ratio
Aritmético: Arithmetic
Nota: O código LZ-78 não se utiliza da estatística da fonte, é baseado em dicionário (adaptativo): Note: The LZ-78 code does not use source statistics, it is dictionary-based (adaptive)

5 CONCLUSION

The general objective of developing and making available tools for consolidating knowledge of Information Theory and Coding was fully achieved. During the work to achieve the general objective, the specific objectives listed in the Objectives section of this document were fully developed. Thus, the non-conformities mentioned in the Results section are already being worked on and the project can be considered concluded.

REFERENCES

Cavalcante, P. C. S. (2021) “ Teoria da Informação e Codificação Notas de Aula”, Disponível em: https://drive.google.com/file/d/1yrLavmXmpQYf-dGVspUwtEWOfAGqsGcB/view?usp=sharing, Acesso em: 14 set. 2021.

INPE (2020) – Instituto Nacional de Pesquisas Espaciais “Vídeo SNCT – Minicurso: Matemática com Python Usando o Mumpy, Matplotlib e Scipy”, Disponível em: https://youtu.be/_iTYsURupgE, Acesso em: 14 set. 2021.

Oliveira,W. C. e Cavalcante, P. C. S. (2021a) “Listagem de Códigos dos Módulos Python desenvolvidos durante a Pesquisa”. Disponível em: https://drive.google.com/file/d/16azruxrezcq52pi2I4au1wxUeNjuKRay/view?usp=sharing, Acesso em: 15 set. 2021.

Oliveira,W. C.e Cavalcante, P. C. S. (2021b) “Resultados dos testes dos Módulos Python desenvolvidos durante a Pesquisa”. Disponível em: https://drive.google.com/file/d/1aXyyPJp3K-vLqXROqKk7XvdSHKqJXh82/view?usp=sharing, Acesso em: 15 set. 2021.

Python Software Foundation (2021) “Documentação do Python”, Disponível em: ttps://docs.python.org/pt-br/3.9/index.html, Acesso em: 14 set. 2021.