

Pesquisa acadêmica para implementação de programas em linguagem Python para medidas de informação e codificação para compressão de dados

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

DOI:10.34117/bjdv8n8-091

Recebimento dos originais: 21/06/2022

Aceitação para publicação: 29/07/2022

Paulo Cesar de Souza Cavalcante

Mestre em Engenharia Elétrica em Telecomunicações pela Universidade Federal de Pernambuco (UFPE)

Instituição: Escola Superior de Tecnologia (EST) da Universidade do Estado do Amazonas (UEA)

Endereço: Av. Darcy Vargas, 1200, Parque 10 de Novembro, Manaus – AM, Brasil, CEP: 69050-020

E-mail: pscavalcante@gmail.com

Williams Cavalcante de Oliveira

Graduando do Curso de Engenharia Eletrônica

Instituição: Escola Superior de Tecnologia (EST) da Universidade do Estado do Amazonas (UEA)

Endereço: Av. Darcy Vargas, 1200, Parque 10 de Novembro, Manaus – AM, Brasil, CEP: 69050-020

E-mail: wco.ele18@uea.edu.br

RESUMO

Este artigo apresenta o resumo expandido contendo informações sobre a pesquisa acadêmica para implementação de programas em linguagem Python para medidas de informação e codificação para compressão de dados. A pesquisa foi realizada, na Universidade do Estado do Amazonas, pelo aluno do curso de Engenharia Eletrônica Williams Cavalcante Oliveira, sob orientação do Professor Paulo Cesar de Souza Cavalcante, e teve seu resumo expandido apresentado na *1st Amazon Stem Academy Conference 2021*. O conteúdo do resumo apresentado, transcrito neste trabalho, expõe breve revisão teórica dos métodos de cálculo de medida de informação e métodos para codificação para compressão de dados baseados em estatística e baseados em dicionários. Além de outras informações concernentes à pesquisa, também apresenta, como seu produto final, os módulos em linguagem Python desenvolvidos para execução dos algoritmos dos métodos estudados, para realização dos cálculos e codificações objetos da pesquisa.

Palavras-chave: medida de informação, codificação para compressão de dados, linguagem Python.

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 Electronic Engineering student 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. In addition to 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 codifications that are the objects of the research.

Keywords: information measurement, coding for data compression, Python language.

1 INTRODUÇÃO

A presente pesquisa teve como objetivo o desenvolvimento de módulos em linguagem Python para realização de cálculos de medida da informação e para compressão de dados, abordados em parte do conteúdo da disciplina Teoria da Informação e Codificação (TIC). Para tanto, foi realizada revisão bibliográfica no material didático de TIC e no material para aprendizagem básica da linguagem de programação Python. Foram desenvolvidos módulos de programa para cálculos de medidas da informação e módulos para implementação dos algoritmos de compressão, baseados em estatística: Shannon-Fano, Huffman, Aritmético, e baseado em dicionários: Lempel Ziv. Os módulos desenvolvidos permitirão que os alunos de TIC, ou outros interessados, verifiquem na prática suas realizações teóricas das funções necessárias para comunicação digital em canais ruidosos ou para sua armazenagem de forma compactada.

2 OBJETIVOS

O objetivo geral do projeto é a criação e disponibilização de ferramentas adicionais para consolidação de conhecimentos durante o curso da disciplina de Teoria da Informação. Os objetivos específicos, contribuintes para o atingimento do objetivo geral são: exercício de trabalho em equipe; aprendizado da linguagem de programação Python; consolidação dos conhecimentos didáticos transmitidos sobre medidas da informação e compressão de dados.

3 MATERIAIS E MÉTODOS

O projeto foi desenvolvido em cumprimento das seguintes etapas: 1) indicação de pelo Orientador de bibliografia básica para pesquisa; 2) realização de pesquisa pelo

Orientando em documentação física e material disponível na Internet; 3) elaboração pelo Orientando de breve referencial teórico sobre Medida da Informação e compressão de dados utilizando seguintes codificações: Shannon-Fano, Huffman, Aritmético e Lempel-Ziv; 4) Implementação pelo Orientando de módulos em linguagem Python para realização de cálculos de Medida da Informação e codificação utilizando as codificações especificadas. 5) Avaliação e orientação pelo Orientador em cada um dos temas especificados na terceira etapa, a medida em que eram concluídas; 6) Elaboração conjunta pelo Orientador e Orientando do presente Resumo Expandido, e 7) Elaboração pelo Orientando, com supervisão do Orientador, da Apresentação do Projeto em conferência.

A bibliografia indicada pelo Orientador na primeira etapa foi: Notas de Aula de Teoria da Informação [Cavalcante 2021]; vídeo de minicurso Matemática com Python [INPE 2020]; site para acesso a Documentação do Python [Python Software Foundation 2021].

3.1 FUNDAMENTAÇÃO TEÓRICA

Nesta subseção será exposta a base teórica sobre Medidas de Informação e Codificação para Compressão de Dados digitais, ou seja dados que são expressos utilizando-se a base de numeração binária, onde só existem dois símbolos: 0 e 1. O material a seguir exposto foi extraído das Notas de Aula de Teoria da Informação [Cavalcante 2021].

I) Medida de Informação

A Informação é tudo aquilo que é produzido por uma fonte para ser transferido ao utilizador. No que diz respeito à medida de informação pode-se considerar dois pontos de vista: do Utilizador, a medida da informação está relacionada com a incerteza (em relação à mensagem que foi transmitida), e da Fonte, a medida da informação é uma indicação da liberdade de escolha exercida pela fonte ao selecionar uma mensagem. Se a fonte possuir muitas mensagens diferentes: a probabilidade de cada mensagem tende a diminuir com o número de mensagens. O utilizador terá mais dúvidas (mais incerteza, mais informação) em relação à mensagem que vai ser escolhida. Por exemplo, se a fonte possuir somente uma mensagem possível: A probabilidade será máxima, igual a 1. O utilizador não terá dúvida (não há incerteza, não há informação) em relação à mensagem que vai ser escolhida.

A) *Medida de Informação de Hartley ou Auto-informação*

Hartley (1928) propôs como medida da quantidade de informação provida pela

observação de uma variável aleatória discreta X . A informação de um único símbolo, $I(X)$, é sugerida por $\log_b K$, onde K é o número de possíveis valores de X . A probabilidade de ocorrência de um dos possíveis valores de X é $P_X=1/K$. Assim, $K=1/P_X$, ou seja K é o inverso da probabilidade de ocorrência de um símbolo. A medida de informação de Hartley:

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

Assim a medida de informação de Hartley pode ser definida como uma grandeza logarítmica ligada ao inverso da probabilidade de um evento. A base do logaritmo usada, define a unidade da medida de informação. Se utilizada a base 2, caso em comunicações digitais, a unidade será bit. Para exemplificar, sejam as probabilidades de emissão por uma fonte dos bits 0 e 1, respectivamente: $P_0 = 1/4$; $P_1 = 3/4$, temos: Informação transportada pelo dígito 0 : $I_0 = -\log_2 1/4 = 2$ bits e Informação transportada pelo dígito 1: $I_1 = -\log_2 3/4 = 0,41$ bits.

B) Medida de Informação de Shannon ou Entropia

Shannon (1948) definiu que em geral, se o i -ésimo valor de X tem probabilidade $P_X(x_i)$, então a informação de Hartley $\log 1/P_X(x_i) = -\log P_X(x_i)$ para este valor deveria ser ponderada por $P_X(x_i)$, fornecendo:

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

A medida de Shannon poderia ser considerada como informação média de Hartley. Shannon chamou esta medida de informação de entropia. A entropia de uma fonte significa que, em média esperamos obter H bits de informação por símbolo. Reescrevendo a fórmula da entropia como:

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

Para exemplificar, supondo-se que uma fonte X emita quatro símbolos x_0, x_1, x_2 e x_3 com probabilidades $1/2, 1/4, 1/8$ e $1/8$, respectivamente. A incerteza, ou entropia $H(X)$

é dada por: $H(X) = (1/2) \log 2 + (1/4) \log 4 + (1/8) \log 8 + (1/8) \log 8 = 1,75$ bits/símbolo.

II) Codificação para Compressão de Dados

O processo de compressão de dados é realizado por algoritmos que recebem M mensagens, sequências de bits de comprimento N , e as codificam para transmissão ou armazenamento em M mensagens, cujo comprimento é menor que N bits das mensagens originais, sem que haja perda das informações (Técnicas de compressão sem perdas). Os modelos estatísticos, ou de codificação por entropia, precisam conhecer a estatística de ocorrência dos símbolos a serem codificados. Os modelos adaptativos, ou baseados em dicionário, executam a compressão sem necessitarem da estatística da fonte. As técnicas abordadas na presente pesquisa se encontram expostas no Quadro 1 abaixo.

Quadro 1. Técnicas de Compressão sem Perdas

Exemplos de Modelos estatísticos	Exemplos de Técnicas baseadas em dicionários
Códigos: Shannon-Fano, Huffman e Aritmética	Códigos Lempel-Ziv(LZ): Lempel-Ziv (LZ 77), Lempel-Ziv (LZ 78) e Lempel-Ziv-Welch (LZW)

O comprimento médio das palavras-código resultantes da codificação para compressão de dados é determinado por:

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

onde l_i é o número de bits da palavra código correspondente ao símbolo i , que ocorre com probabilidade p_i . A eficiência de codificação pode ser definida pela relação entre a entropia da fonte ou da mensagem e o comprimento médio das palavras de código:

$$\eta = \frac{H(S)}{\bar{L}} \quad (5)$$

A taxa de compressão T_c de um código compressor é definida pela expressão abaixo:

$$T_c = \frac{Q_{\text{de bits texto sem compressão}} - Q_{\text{de bits texto com compressão}}}{Q_{\text{de bits texto sem compressão}}} \times 100 \quad (6)$$

3.2 DESENVOLVIMENTO DOS MÓDULOS EM PYTHON

O desenvolvimento dos módulos em linguagem Python foram realizados na plataforma *Intergrated Development Environment* (IDE) PyCharm, em produção em plataforma computacional com sistema operacional Windows 10. Foram implementados os seguintes módulos Python para cálculo da entropia e compressão utilizando a codificação pelos respectivos algoritmos: Entropia; Shannon-Fano; Huffman; Aritmético; Lempel-Ziv LZ 78. Os módulos desenvolvidos se encontram no documento Listagens dos módulos Python [Oliveira e Cavalcante 2021a].

4 RESULTADOS

Foram realizados os testes de compressão de mensagens pelos módulos Python desenvolvidos. Os Quadro 2 e Quadro 3 seguintes apresentam uma consolidação dos dados extraídos da execução dos módulos, constantes do documento Resultados dos testes dos módulos Python [Oliveira e Cavalcante 2021b]. Os dados expostos são a própria codificação e parâmetros de medidas expostos na fundamentação teórica deste trabalho.

Quadro 2. Teste de compressão da Mensagem em ASCII (8 bits/ simb.): DEMONSTRAÇÃO PARA FIRST AMAZON STEM ACADEMY CONFERENCE

Símbolos texto			Código Shannon Fano			Código Huffman			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
									'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
Texto sem codificação			Código	Comp.	Eficiência	Taxa					
				Médio	Código	Compressão					
				(Form. 4)	(Form.5)	(Form. 6)					
Total de bits			Shannon	3,9 b/s	99,21%						
54 x 8 bits/s= 432 bits			Huffman	3,9 b/s	99,21%	51,16%					
Entropia do texto			LZ78	11 b/s	35,18%	31,25%					
(Fórmula 3)			Nota: O código LZ-78 não se utiliza da estatística da								
H(P) = 3,87 bis/s			fonte, é baseado em dicionário (adaptativo)								

Embora não tenham ocorrido erros nos testes de compressão realizados conforme Quadros 2 e 3, em outros testes foram verificadas algumas inconformidades nos módulos de Entropia (totalização de probabilidades), Aritmético (erro de conversão valor decimal para binário) e Lempel-Ziv LZ-78 (supressão da codificação do último segmento de mensagem), as quais já se encontram em análise para eliminação.

Quadro 3. Teste de compressão da Mensagem em ASCII (8 bits/ simb.): asadacasa

Símbolos 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				Código	Comp.	Eficiência	Taxa
9 x 8 bits/s= 72 bits					Médio	Código	Compressão
					(Form. 4)	(Form.5)	(Form. 6)
Entropia do texto				Aritmético	10 b/s	-	86,11%
(Fórmula 3)				LZ78	6 b/s	-	51,16%
H(P) = 1,65 bis/s				Nota: O código LZ-78 não se utiliza da estatística da fonte, é baseado em dicionário (adaptativo)			

5 CONCLUSÃO

O objetivo geral de desenvolvimento e disponibilização de ferramentas para consolidação de conhecimentos de Teoria da Informação e Codificação foi alcançado em toda sua plenitude. Durante os trabalhos para atingimento do objetivo geral os objetivos específicos listados na seção Objetivos deste documento foram plenamente desenvolvidos. Assim, as inconformidades citadas na seção de Resultados já estão sendo trabalhadas e o projeto pode ser considerado concluído.

REFERÊNCIAS

Cavalcante, P. C. S. (2021) “ Teoria da Informação e Codificação Notas de Aula”, Disponível em: <https://drive.google.com/file/d/1yrLavmXmpQYfdGVspUwtEWOofAGqsGcB/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: <https://docs.python.org/pt-br/3.9/index.html>, Acesso em: 14 set. 2021.