UNIVERSIDADE D
COIMBRA

João Botelho, Nº 2019155348, uc2019155348@student.uc.pt
Guilherme Branco, Nº 2020216924, mbranco@student.dei.uc.pt
Dário Félix, Nº 2018275530, dario@student.dei.uc.pt

TP1
Entropia, Redundância e Informação Mútua

Relatório no âmbito da cadeira de Teoria da Informação da Licenciatura em Engenharia Informática, orientado pelo Professor Rui Paiva (PL2), do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Novembro de 2021

Índice

Introdução

Este trabalho tem como objetivo consolidar os conhecimentos na área da teoria da informação, nomeadamente conceitos como a entropia, códigos de Huffman e informação mútua.

Para isso, serão utilizados ficheiros de áudio, imagem e texto como fontes de informação para determinar o histograma de ocorrências dos seus símbolos, calcular a entropia (com e sem agrupamento de símbolos), determinar o número médio de bits por símbolo usando a codificação de Huffman (além da variância dos comprimentos desses códigos) e, por fim, calcular a informação mútua entre um sinal a pesquisar (*query*) e um sinal onde pesquisar (*target*).

Neste documento é abordada a implementação do código, a exposição dos resultados obtidos e, por fim, a discussão e a análise desses resultados, para cada alínea do enunciado.

O trabalho foi desenvolvido em Python 3.8.8.

Palavras-chave

Histograma de ocorrência dos símbolos; Informação; Redundância; Entropia; Códigos de Huffman; Variância dos comprimentos dos códigos de Huffman; Agrupamento de símbolos; Informação mútua; Entropia conjunta.

1. Implementação e Metodologia

Nesta secção é abordada a implementação das funções e das escolhas feitas na construção dos algoritmos para, por exemplo, minimizar a complexidade temporal ou para torná-las mais genéricas.

Todo o código elaborado está no ficheiro "*mainTP1.py*". Esse ficheiro contém uma função chamada *main()*, que tem como objetivo ler todas as fontes de informação fornecidas (áudio, imagem e texto), compatibilizá-las com as funções desenvolvidas, e executar estas funções com os parâmetros necessários.

1.1. Alínea 1

A função *histograma(data, fname)* extrai o alfabeto e contagem de ocorrências a partir da fonte fazendo uso de *alphabet(data)* ou *alphabet_text(data)*, consoante esta fonte contenha dados numéricos ou texto, respetivamente. Em seguida, usando a função *bar()* do *matplotlib,* reproduz no ecrã o histograma das ocorrências de cada símbolo do alfabeto.

1.2. Alínea 2

A entropia é calculada segundo a seguinte fórmula [1]:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

A função *entropia(lista_ocorrencias)* é facilmente construída: dado um *array* com a contagem das ocorrências de cada símbolo do alfabeto (*lista_ocorrencias*), e utilizando os métodos do *numpy,* determina-se o *array* das probabilidades de ocorrência de cada símbolo (*lista_prob*), dividindo a *lista_ocorrencias* pelo somatório das ocorrências da *lista_ocorrencias*. Por fim, aplica-se o somatório da multiplicação da *lista_prob* pelo logaritmo de base 2 de 1 sobre a *lista_prob*.

1.3. Alínea 4

A função *huffman_media_variancia(data, lista_ocorrencias)* calcula a média e a variância dos códigos resultantes das funções de codificação de *Huffman* que são fornecidas em "*huffmancodec.py*".

Dado um *array* com os dados de uma fonte de informação (*data*), extraem-se os comprimentos dos códigos (*lenghts*), utilizando as funções de codificação. À semelhança da alínea 2, dado um *array* com a contagem das

ocorrências de cada símbolo do alfabeto (*lista_ocorrencias*), determina-se o *array* das probabilidades de ocorrência de cada símbolo (*lista_prob*), dividindo a *lista_ocorrencias* pelo somatório das ocorrências da *lista_ocorrencias*. Depois, são retirados todos os elementos nulos da *lista_prob*.

Tanto a média como a variância são valores ponderados e calculados utilizando o método *average()* do *numpy*. Esse método recebe os *lengths* para o cálculo da média e (*lengths* – média)^2 para o cálculo da variância. Em ambos os casos, recebe também *lista_prob* para definir o peso de cada elemento).

1.4. Alínea 5

Dado um *array* de dimensão 1xN, a função *grouping(data)* faz o agrupamento dos símbolos transformando-o num *array* 2x(N/2), calcula a entropia dos símbolos agora agrupados e, por se tratar de dois valores por símbolo, divide o valor da entropia por 2.

1.5. Alínea 6

A informação mútua é calculada segundo a seguinte fórmula [1]:

$$I(X;Y) = \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

O cálculo é feito pela função *shazam(query, target, alpha, step)*, onde *query* e *target* são os dos dados a serem comparados, *alpha* é o alfabeto dos dados, e *step* é o passo dado pela janela deslizante, sendo o tamanho da janela o tamanho de *query*.

A cada passo da janela, a função *mutualInfo(query, target, alpha)* calcula os valores de P(x,y), P(x) e P(y), faz a multiplicação de P(x) por P(y), divide P(x,y) por P(x)P(y), e por fim calcula o somatório de P(x,y) multiplicado pelo log base dois de P(x,y)/P(x)P(y), sendo x=*query* e y=*target*. O resultado é a informação mútua.

2. Resultados

Em seguida são apresentados os resultados obtidos.

2.1. Alínea 3

Histogramas do número de ocorrências de cada símbolo para cada fonte de informação (*Figura 1*):

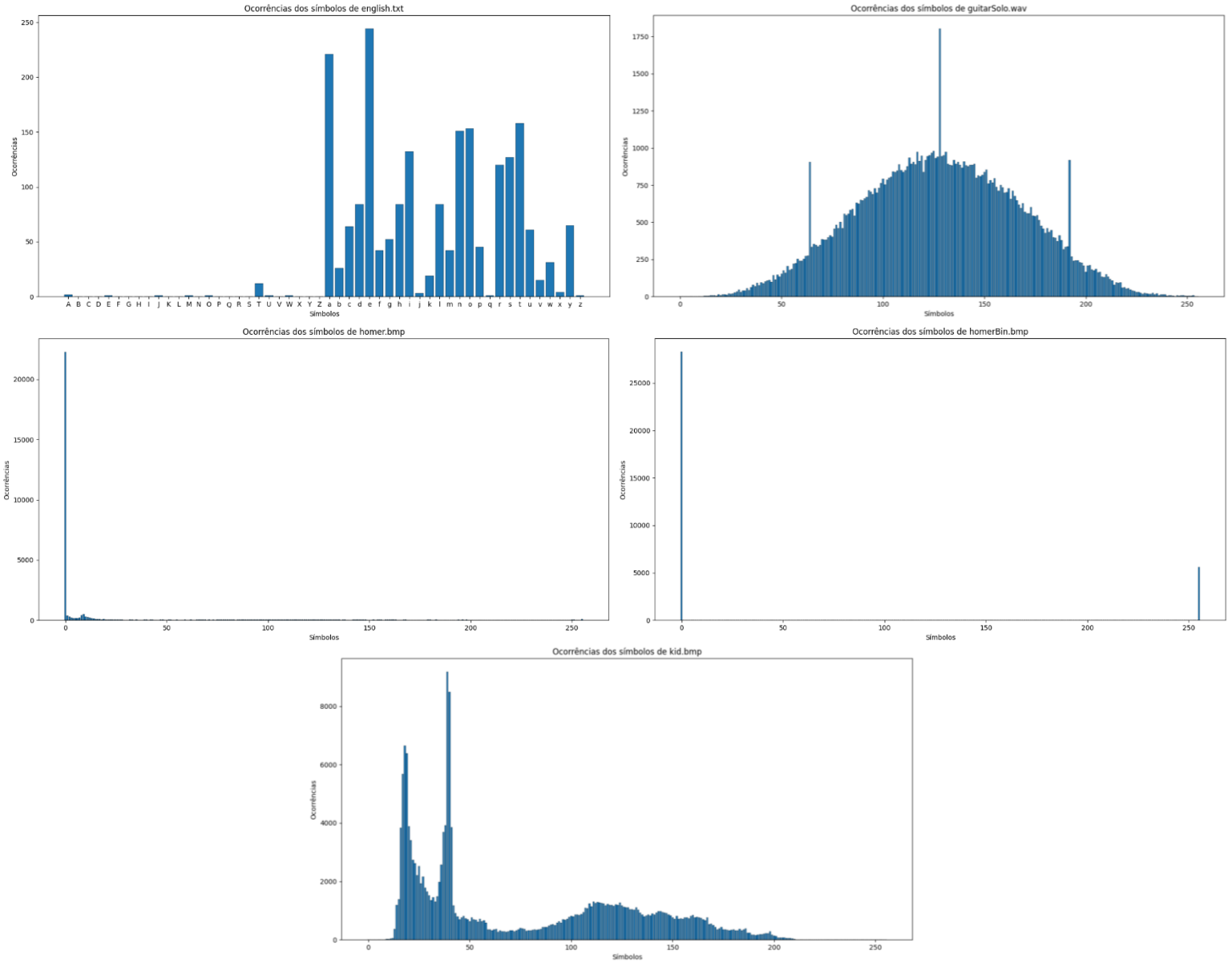

Figura 1 – Histogramas - da esquerda para a direita: english.txt, guitarSolo.wav, homer.bmp, homerBin.bmp e kid.bmp.

Destaca-se que no ficheiro "*homerBin.bmp*" se verifica a ocorrência de apenas dois símbolos. Em "*guitarSolo.wav*" e "*kid.bmp*", por outro lado, há um grande número de ocorrências para um espetro largo dos seus alfabetos.

Limite mínimo para o número médio de bits por símbolo (entropia):

Fonte de informação	Entropia (bits/símbolo)
english.txt	4.228
guitarSolo.wav	7.329
homer.bmp	3.466
homerBin.bmp	0.645
kid.bmp	6.954

2.2. Alínea 4

Resultados do uso da codificação de Huffman:

- Número médio de bits por símbolo:

Fonte de informação	Média (bits/símbolo)
english.txt	4.252
guitarSolo.wav	7.351
homer.bmp	3.548
homerBin.bmp	1.0
kid.bmp	6.983

- Variância dos comprimentos dos códigos:

Fonte de informação	Variância (bits/símbolo)
english.txt	1.191
guitarSolo.wav	0.727
homer.bmp	13.197
homerBin.bmp	0.0
kid.bmp	2.099

2.3. Alínea 5

Limite mínimo para o número médio de bits por símbolo (entropia) com o agrupamento de símbolos 2 a 2:

Fonte de informação	Entropia (bits/símbolo)
english.txt	3.652
guitarSolo.wav	5.754
homer.bmp	2.413
homerBin.bmp	0.398
kid.bmp	4.909

2.4. Alínea 6b

Evolução da informação mútua ao longo do tempo para cada *target* (*Figura 2*):

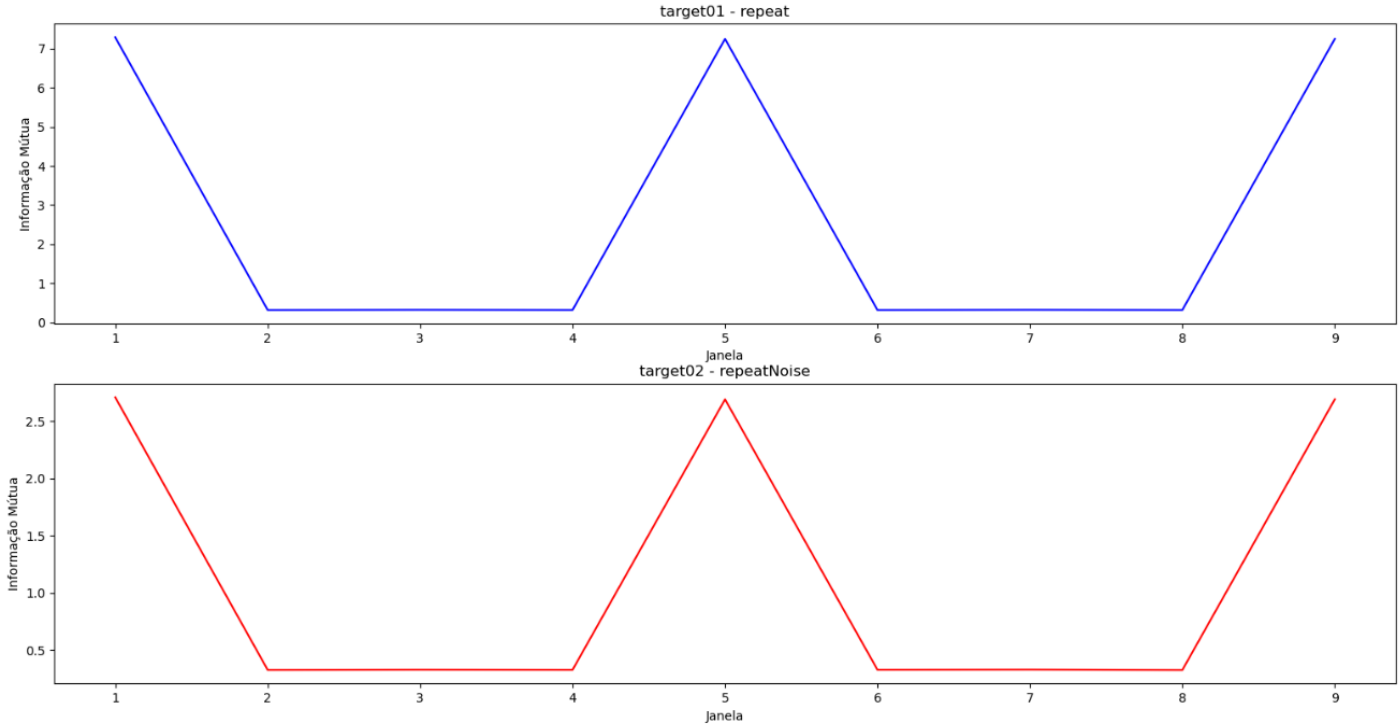

Figura 2 – Evolução da informação mútua; a azul: target01 – repeat.wav; a vermelho: target02 – repeatNoise.wav.

Percebe-se que a variação da informação mútua em ambos os *targets* (*target01 - repeat* e *target02 – repeatNoise*) é semelhante, apesar de apresentar valores diferentes em cada instante.

2.5. Alínea 6c

Informação mútua máxima de "*guitarSolo.wav*" com cada fonte de informação *target*, por ordem decrescente:

Target	Informação mútua
Song06.wav	7.310
Song07.wav	6.297
Song05.wav	3.960
Song04.wav	0.398
Song02.wav	0.367
Song03.wav	0.297
Song01.wav	0.252

3. Discussão

Segue-se a análise e comentário dos resultados obtidos.

3.1. Alínea 3

Verifica-se no histograma do ficheiro "*guitarSolo.wav*" que os seus símbolos têm uma distribuição de probabilidade menos variável, em comparação com os restantes ficheiros: há uma menor diferença entre os números de ocorrências dos seus símbolos. Assim, é expectável que apresente o valor de entropia mais alto, o que se confirma.

No ficheiro "*kid.bmp*", a existência de alguns símbolos com ocorrências muito superiores tem menos impacto na sua entropia devido às altas ocorrências de todos os outros símbolos. Assim, "*kid.bmp*" apresenta também uma entropia elevada.

O ficheiro "*homerBin.bmp*" tem apenas dois símbolos, sendo que um tem uma probabilidade de ocorrência significativamente superior. Isto justifica a sua entropia inferior a 1 (valor que ocorreria se os símbolos fossem equiprováveis).

"Será possível comprimir cada uma das fontes de forma não destrutiva? Se Sim, qual a compressão máxima que se consegue alcançar? Justifique."

Pelo teorema de Shannon, a máxima compressão não destrutiva alcançável é limitada inferiormente pelo valor da entropia. Assim, é possível comprimir as fontes de forma não destrutiva, sendo esta compressão limitada, no máximo, aos números médios de bits por símbolo obtidos. [1]

3.2. Alínea 4

O número médio de bits por símbolo obtido para uma fonte de informação S pela codificação de Huffman é limitado ao intervalo entre a entropia, H(S), e H(S)+1.

Isto é verificado pelos valores obtidos para os ficheiros, que são superiores às entropias calculadas na Alínea 3 mas não a excedem por mais que uma unidade.

A variância do comprimento do código de Huffman obtido para *homerBin.bmp* é 0.0, tal como esperado para um alfabeto de apenas dois símbolos.

Destaca-se também a diferença de variâncias entre "*homer.bmp*" e "*kid.bmp*", apesar de terem alfabetos semelhantes. As ocorrências dos

símbolos de "*kid.bmp*" são relativamente próximas da média, o que se traduz num número de bits para cada símbolo próximo da média. Em "*homer.bmp*", existe um símbolo cujo número de ocorrências é aproximadamente 100 ordens de magnitude superior aos dos restantes, o que leva alguns símbolos a terem códigos muito mais longos do que este. Assim, a variância dos comprimentos dos códigos é elevada.

"Será possível reduzir-se a variância? Se sim, como pode ser feito e em que circunstância será útil?"

É possível reduzir a variância combinando os símbolos e colocando-os na ordem mais elevada possível da árvore de Huffman ao construir o código. A redução desta variância é útil na transmissão de informação: é conveniente que a taxa de enchimento do buffer seja aproximadamente constante. [1]

3.3. Alínea 5

As entropias dos ficheiros diminuem com o agrupamento dos seus símbolos, tal como esperado, pois sabendo o contexto (símbolo anterior), a probabilidade é maior ou menor para um determinado símbolo seguinte, provocando uma diminuição da incerteza (entropia).

3.4. Alínea 6b

A informação mútua entre "*guitarSolo.wav*" e um *target* é máxima quando esta for igual à entropia de "*guitarSolo.wav*". Esse máximo verifica-se em algumas janelas de "*target01 – repeat.wav*". Isto não acontece para "*target02 – repeatNoise.wav*", devido ao ruído aleatório que diminui a informação mútua entre os ficheiros, mas continua a existir uma dependência estatística visível entre este ficheiro e "*guitarSolo.wav*".

3.5. Alínea 6c

O ficheiro "*Song06.wav*" apresenta um valor de informação mútua muito próximo ao valor máximo que pode atingir, visto que sonoramente "*guitarSolo.wav*" é um excerto de "*Song06.wav*".

Os ficheiros "*Song07.wav*" e "*Song05.wav*" apresentam uma informação mútua relativamente intermédia face aos restantes targets. Verifica-se que a intensidade do ruído influencia a informação mútua, diminuindo-a, visto que estes ficheiros contêm a mesma música que o "*Song06.wav*" com intensidades de ruído diferentes. Apesar disso, continua

a existir uma dependência estatística visível e expressiva no valor da informação mútua.

Os ficheiros "*Song04.wav*", "*Song02.wav*", "*Song03.wav*" e "*Song01.wav*" têm valores de informação mútua com "*guitarSolo.wav*" muito baixos e próximos entre si. Estes valores são previsíveis, porque a música de onde "*guitarSolo.wav*" origina é diferente e sem qualquer relação com estes quatro targets.

4. Conclusão

Este trabalho introduziu questões fundamentais de teoria da informação, como a informação, redundância, entropia e informação mútua.

Os objetivos deste trabalho foram cumpridos e não foram sentidas quaisquer dificuldades na sua execução. Para tal, contribuiu o empenho e dedicação de todos os membros do grupo.

Referências

[1] "Cap. II - Teoria da Informação e Codificação Entrópica", slides da cadeira de TI, acedido em novembro de 2021.