FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Relatório
Trabalho Prático nº1

Entropia, Redundância e Informação Mútua

Teoria da Informação

Diogo Cleto - 2019198370
Miguel Pedroso – 2019218176
Thomas Fresco – 2019219057

Coimbra, 27 de outubro de 2020

Introdução

Neste trabalho, pretende-se aprofundar os conhecimentos aprendidos nas aulas de Teoria da Informação, designadamente assuntos relacionados com Entropia, Redundância e Informação Mútua. Utilizar-se-á a linguagem Python e as respetivas bibliotecas adequadas aos exercícios propostos.

O conceito de entropia será a base do trabalho. Entenda-se este conceito como o número médio de bits por símbolo necessários para codificar uma fonte de informação. Quanto maior for a entropia de uma certa fonte, maior é a incerteza associada a essa fonte, e vice-versa.

Ao calcular este valor para uma dada fonte, descobre-se a redundância que lhe está associada, uma vez que a entropia traduz o valor de compressão máxima para um conjunto de dados, sem que haja perdas de informação.

Para cálculo da entropia, utilizar-se-á a seguinte fórmula:

$$H = -\sum p(x) \log p(x)$$

1. No primeiro exercício pretende-se, a partir de uma dada fonte de informação P = {p1, ..., pm} e de um alfabeto A = {a1, ..., an}, determinar e visualizar o histograma de ocorrência dos símbolos de A em P. Para tal, foi desenvolvida a função “histograma” que recebe três parâmetros, o nome de ficheiro a ler (parâmetro relevante para a pergunta 3), a matriz correspondente aos dados da fonte e o alfabeto. No dicionário “leitura”, guarda-se , num lado, os elementos do alfabeto e, do outro (key), o número de ocorrências de cada símbolo. Este dicionário vai ser utilizado de seguida para gerar o gráfico com o alfabeto no eixo xx e as ocorrências no eixo yy.

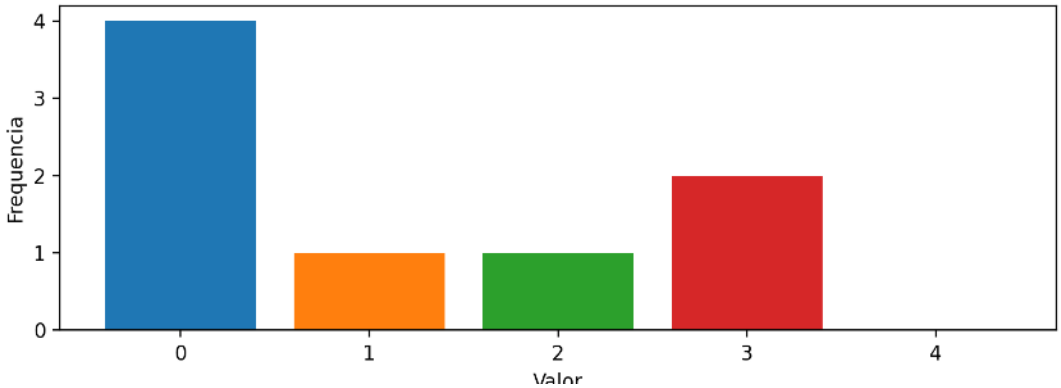

Fig.1 - Gráfico obtido através da interseção A = [0,1,2,3,4] e P = [0,1,3,2,0,0,3,0]

2. Neste ponto foi pedido para construir uma função que, dada uma fonte de informação e um determinado alfabeto, determine a entropia, ou seja, o limite mínimo teórico para o número médio de bits por símbolo.

3. No exercício 3, é pedido, através das funções desenvolvidas em 1 e 2, que se apresente a distribuição estatística da fonte e o número médio de bits por símbolo, ou seja, a entropia, para cada uma das fontes fornecidas: lena.bmp, ct1.bmp, binaria.bmp, saxriff.wav e texto.txt.

Análise dos gráficos e resultados

Nome Ficheiro	Entropia (bits/símbolo)
lena.bmp	6.915336
ct1.bmp	5.972234
binaria.bmp	0.975527
saxriff.wav	3.530989
texto.txt	4.196889

Análise do ficheiro lena.bmp

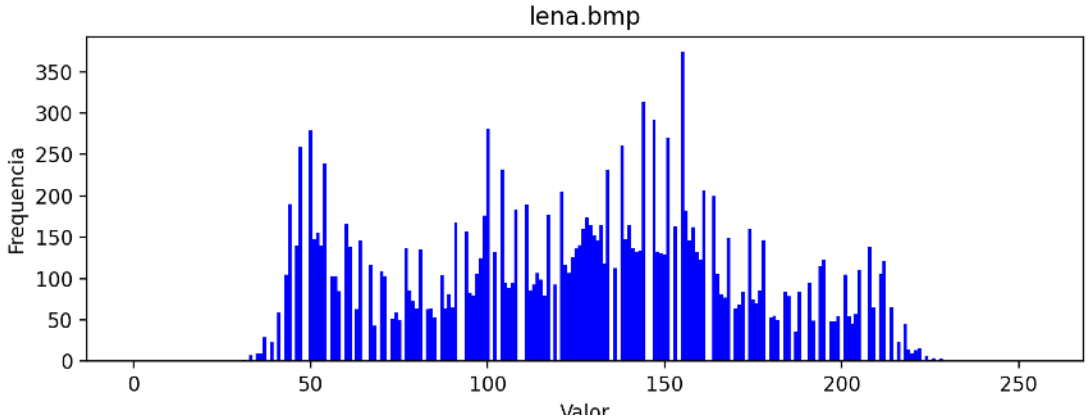

Fig.2 - Gráfico obtido por lena.bmp

Nesta imagem pode-se observar um número elevado de cores, mais concretamente, níveis de cinzento, o que, logicamente, provoca uma entropia elevada. Tal é comprovado ao observar o gráfico, verificando-se uma grande distribuição dos valores, o que implica uma maior incerteza na leitura e por consequência uma maior entropia.

Análise do ficheiro ct1.bmp

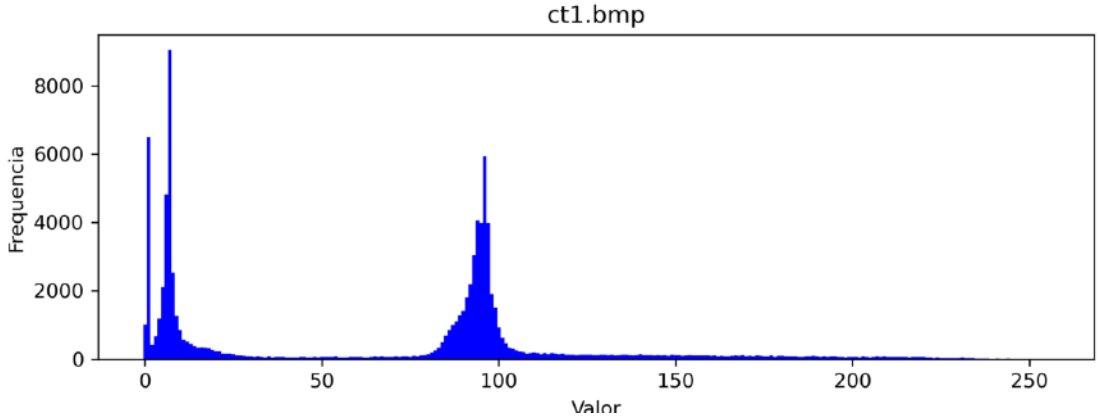

Fig.3 - Gráfico obtido por ct1.bmp

Tal como em “lena.bmp”, ao analisar a imagem, verifica-se a existência de algumas cores, mas em menor diversidade e mais aproximadas em valores do que na imagem anterior. Isto significa que a entropia deste será, com certeza, menor. Conclui-se, assim, que se obtém uma distribuição estatística mais “compacta”, o que implica uma menor incerteza e uma menor entropia, quando comparado com o ficheiro anterior.

Análise do ficheiro binaria.bmp

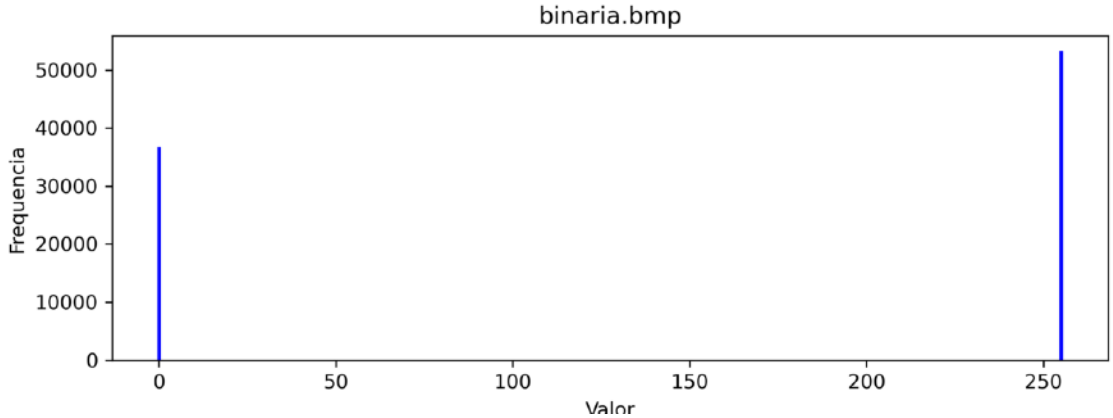

Fig.4 - Gráfico obtido por binaria.bmp

Nesta imagem, pode-se, desde logo, prever uma baixa entropia, uma vez que se trata de uma imagem binária, ou seja, só possui duas cores, preto e branco. Observa-se esse facto no gráfico, com a existência de apenas dois valores: 0 (preto) e 255 (branco). Como só existem dois valores possíveis, a incerteza é muito baixa, resultando no menor valor de entropia até agora observado (0.975527 bits/símbolo).

Análise do ficheiro saxriff.wav

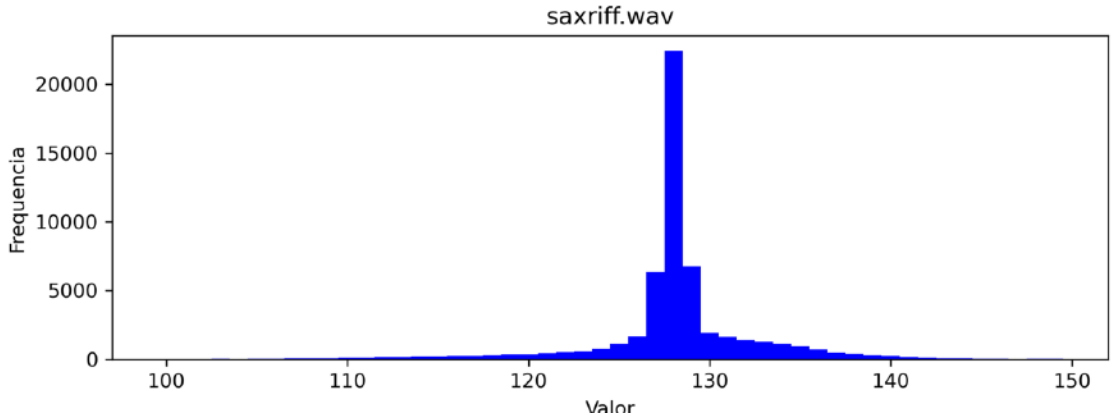

Fig.5 - Gráfico obtido por saxriff.wav

Ao observar o gráfico do ficheiro de áudio, verifica-se uma distribuição de valores bastante diferente da das imagens. É de notar uma elevada frequência nos valores do meio, o que, numa interpretação mais científica, traduz a passagem da onda sonora pela posição de equilíbrio, posição 0 em termos físicos. Tal resulta na segunda menor entropia até agora registada.

Análise do ficheiro texto.txt

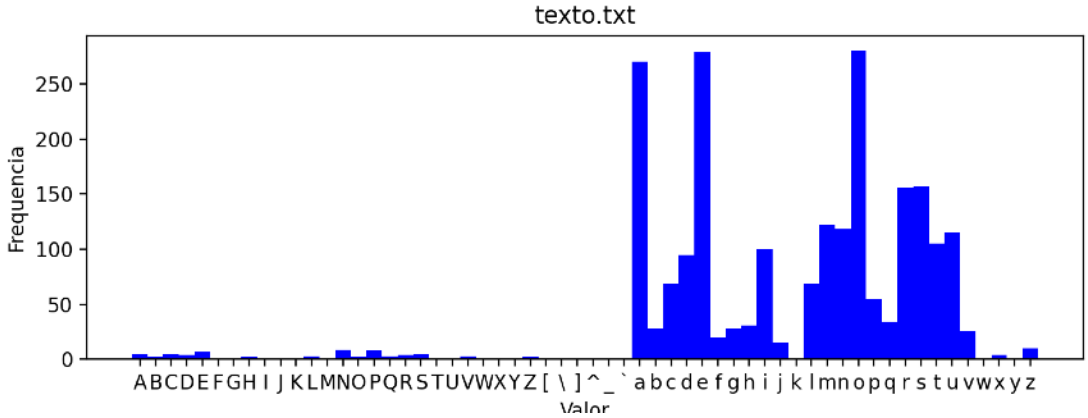

Fig.6 - Gráfico obtido por texto.txt

No ficheiro de texto, observa-se uma maior frequência de letras minúsculas (valores entre 97 e 122) do que letras maiúsculas (valores entre 65 e 90). Como os valores possíveis são poucos, a incerteza não é muita e a entropia é de 4.196889 bits por símbolo.

Será possível comprimir cada uma das fontes de forma não destrutiva?

Para um ficheiro, o valor de compressão não destrutiva máxima (%) é dado por:

(Entropia Máxima-Entropia)/Entropia Máxima*100

Entropia Máxima - número de bits necessário para codificar cada símbolo.

Para cada uma das fontes de informação fornecidas, temos:

- lena.bmp: ((8-6.915336)/8) * 100 = 13,56%
- ct1.bmp: ((8-5.972234)/8) * 100 = 25,35%
- binaria.bmp: ((8-0.975527)/8) * 100 = 87,81%
- saxriff.wav: ((8-3.530989)/8) * 100 = 55,86%
- texto.txt: ((8-4.196889)/8) * 100 = 47,54%

Podemos concluir que sim, é possível comprimir cada uma das fontes de uma forma não destrutiva.

4.

Nome Ficheiro	Huffman (bits/símbolo)	Variância
lena.bmp	6.942566	0.640439
ct1.bmp	6.007558	5.201886
binaria.bmp	1.408164	0.241566
saxriff.wav	3.441179	6.913860
texto.txt	4.217738	1.889175

Na questão 4, é pedido, utilizando o código Huffman fornecido, o cálculo do número médio de bits por símbolo para cada uma das fontes de informação. Observa-se que os novos valores se distanciam ligeiramente do valor mínimo teórico para o número médio de bits por símbolo, calculado na pergunta anterior.

Ao observar a variância, verifica-se uma elevada dispersão do número de bits necessários para codificar os diferentes símbolos. Uma forma de reduzir esta dispersão passa por colocar todos os símbolos na ordem mais alta permitida na árvore, de acordo com a sua ocorrência. Assim, cada símbolo passará a ser codificado pelo menor número de bits possível.

5. Neste exercício, a informação de cada fonte foi organizada de forma diferente, resultando no agrupamento dos símbolos dois a dois. De seguida, foi calculada a entropia para cada ficheiro, tendo em conta esta nova condição.

Nome Ficheiro	Entropia agrupados 2 a 2 (bits/símbolo)	Entropia sem agrupamento (bits/símbolo)
lena.bmp	5.596516	6.915336
ct1.bmp	4.481268	5.972234
binaria.bmp	0.542405	0.975527
saxriff.wav	2.889826	3.530989
texto.txt	3.752924	4.196889

Como seria expectável, ao agrupar os símbolos dois a dois, a entropia de cada fonte de informação é menor, quando comparada com a entropia calculada inicialmente (sem agrupamento de símbolos). De forma geral, o agrupamento de símbolos permite incertezas menores, o que resulta numa entropia reduzida.

6.

a) Nesta alínea, é pedido o cálculo da informação mútua para um par de fontes, exercendo uma a função de *query e a outra,* a função de *target.* Para isto, é criada uma "janela deslizante" que faz com que a *query* vá percorrendo o *target* com um determinado passo, valor também a definir. O nº de janelas vai ser o mesmo nº de excertos diferentes do *target* que vão ser comparados diretamente com a *query*, calculando assim um valor de informação mútua para cada análise. Essa informação é guardada/organizada num *array*.

b) Utilizando o novo código implementado na alínea anterior, é criada e adaptada uma nova função para receber uma *query*, um *target*, um passo e um alfabeto adequado, articulando a resolução desta alínea com as primeiras funções desenvolvidas neste trabalho.

Foi proposta a utilização o ficheiro "saxriff.wav" como *query* para determinar a informação mútua com os ficheiros "target01 - repeat.wav" e "target02 – repeatNoise.wav", que servirão de *target*. Os resultados obtidos encontram-se representados nos seguintes gráficos:

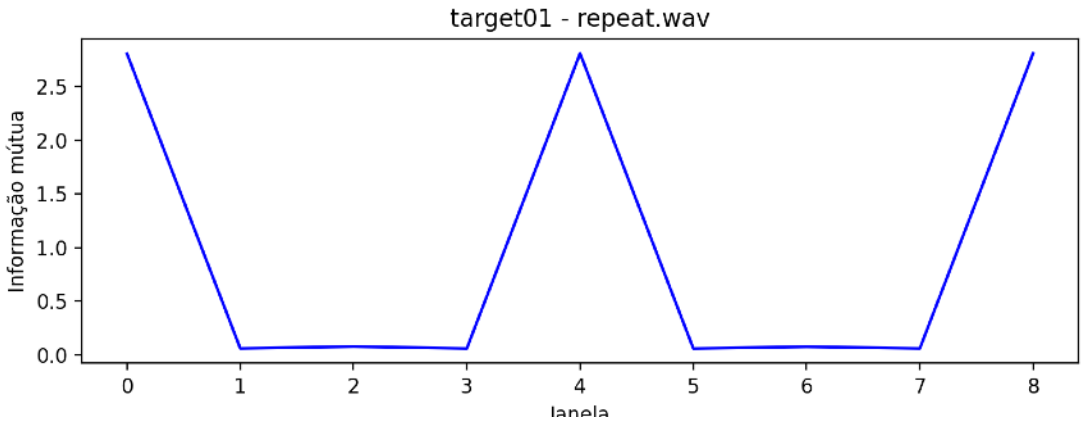

Fig.7 - Gráfico informação mútua para o target nº 1

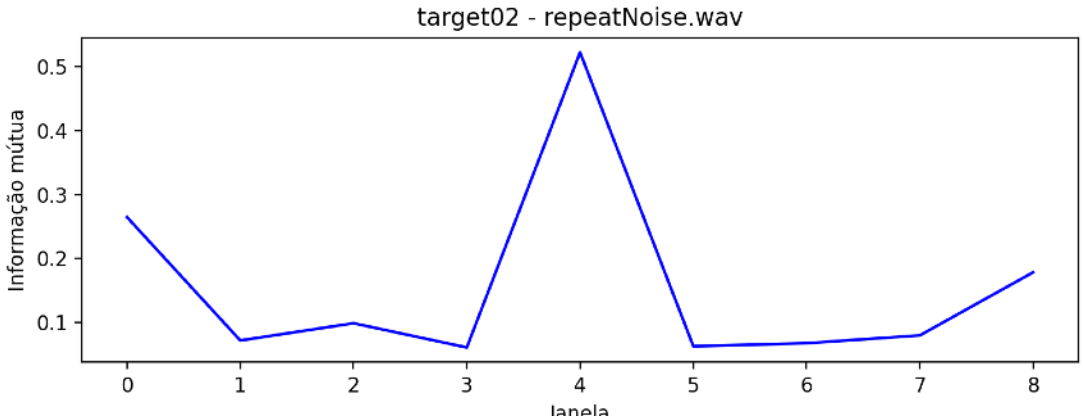

Fig.8 - Gráfico informação mútua para o target nº 2

Analisando os gráficos, pode-se concluir que a informação mútua é, no geral, maior entre "saxriff.wav" e "target01 - repeat.wav" do que entre "saxriff.wav" e "target02 - repeatNoise.wav". Verifica-se tal facto pois "target01 - repeat.wav" resulta apenas de repetições sucessivas de "saxriff.wav" (3 vezes). "target02 - repeatNoise.wav" é um ficheiro semelhante ao "target01 - repeat.wav", mas com a adição de algum ruído, o que dificulta o processo de descodificação e provoca uma diminuição dos valores de informação mutua.

c) O objetivo desta última alínea é criar um pequeno simulador de identificação de música, utilizando novamente o 'saxriff.wav' como *query* e 7 *targets* diferentes, determinando a informação mútua máxima entre a *query* e cada um destes *targets*.

Nome Ficheiros	Informação Mútua (Máximo)
saxriff.wav e song06.wav	3.53098873
saxriff.wav e song07.wav	3.53098873
saxriff.wav e song05.wav	0.53224241
saxriff.wav e song04.wav	0.19468622
saxriff.wav e song02.wav	0.18903791
saxriff.wav e song03.wav	0.16715209
saxriff.wav e song01.wav	0.13634246

Conclui-se, por leitura direta dos dados presentes na tabela acima, que os *targets* “song06.wav” e “song07.wav” são os que têm maior compatibilidade com a query “saxriff.wav”, visto que que são os ficheiros que traduziram o valor de informação mútua mais elevado, com o valor máximo de 3.53098873. Pelo contrário, pode-se afirmar que a *query* em estudo tem menor a compatibilidade com “song01.wav”, com o valor máximo de 0.13634246 para a informação mútua.

Conclusão

Em conclusão, foram atingidos os objetivos do trabalho, na medida em que se aprofundaram conhecimentos relacionados com Entropia, Redundância e Informação Mútua, bem como o domínio da linguagem Python e das bibliotecas utilizadas.

Através do cálculo da entropia e da perceção do conceito, foi possível provar algumas hipóteses, como por exemplo, que é possível comprimir diversas fontes de informação uma forma não destrutiva. Depois, este estudo permitiu analisar de forma crítica a variância obtida para cada grupo de dados e, também, os valores de entropia para situações em que os símbolos foram agrupados 2 a 2, entre outros aspetos.

Por último, após o domínio dos conteúdos anteriores, partiu-se para a criação de um pequeno simulador de identificação de música, onde foi introduzido o conceito de informação mútua e ponderada a semelhança entre diferentes ficheiros áudio (relação *query-target*).