

Relatório do 1º trabalho teórico- prático

DEI – Departamento de Engenharia
Informática - FCTUC

Estudantes:

Gonçalo Fernandes Diogo de Almeida, 2020218868

João Bernardo de Jesus Santos, 2020218995

Samuel Eduardo Brinca Machado, 2020219391

Professores:

Paulo Carvalho (T1 e TP1) e Marco Simões (PL4)

Disciplina:

Teoria da Informação

Curso – Ano:

Licenciatura em Engenharia Informática – 2021/2022

Índice

Conteúdo

Objetivo.....	3
Exercícios	3
Exercícios 1 e 2.....	3
Exercício 3	3
Exercício 4	6
Exercício 5	7
Exercício 6	7

Objetivo

Pretende-se que adquiramos sensibilidade para as questões fundamentais de teoria de informação, em particular informação, redundância, entropia e informação mútua.

Exercícios

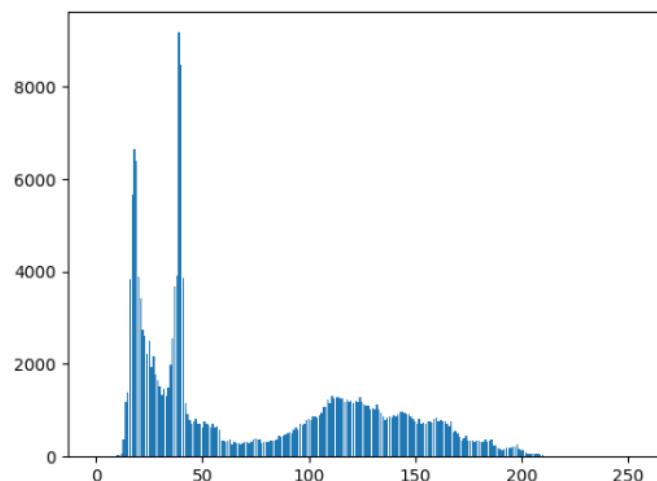
Exercícios 1 e 2

Nestes 2 primeiros exercícios é-nos pedido, dada uma fonte de informação P com um alfabeto $A = \{a_1, \dots, a_n\}$, para determinar e visualizar o histograma de ocorrência dos seus símbolos (Exercício 1) e determinar o limite mínimo teórico para o número médio de bits por símbolo (Exercício 2). Para isso, no exercício 1, criámos a função 'hist' que mostra as ocorrências referidas anteriormente. No exercício 2, criámos duas funções: a 'prob', que devolve uma tabela com as probabilidades de ocorrência de cada elemento da tabela passada como argumento; e a função 'entropia', que devolve o valor da entropia de uma tabela de probabilidades passada como argumento.

Exercício 3

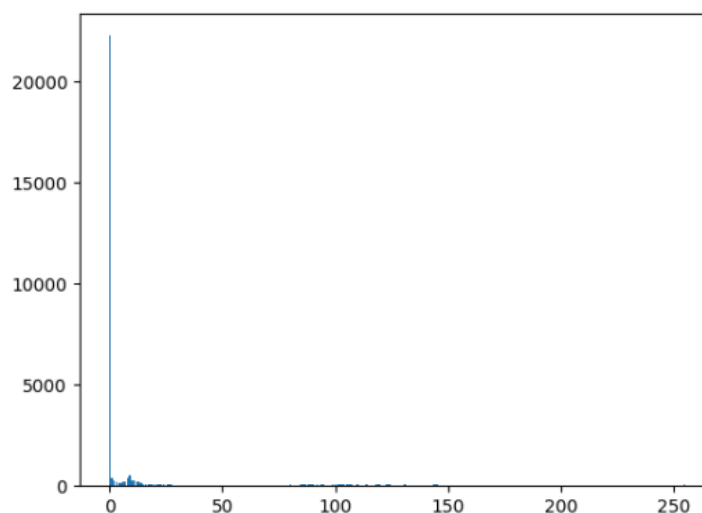
No exercício 3, é-nos pedido para determinar a distribuição estatística e o limite mínimo para o número médio de bits por símbolo das fontes fornecidas.

Fonte: kid.bmp



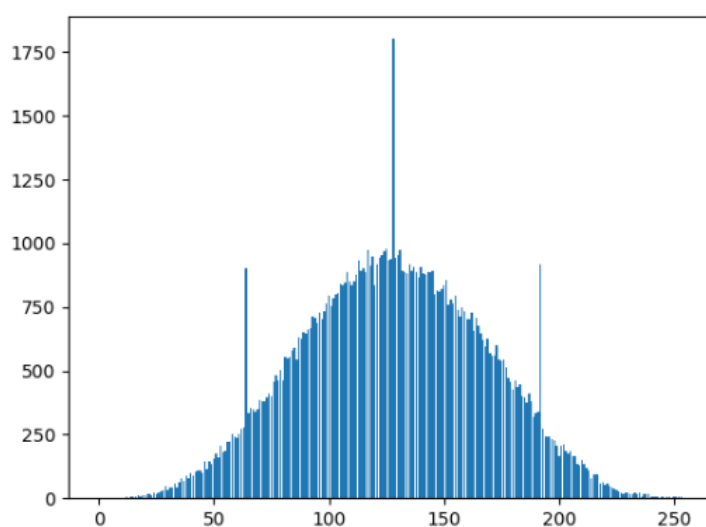
Trata-se de uma imagem a preto e branco, portanto os tons de cinza (incluindo o preto e o branco) presentes na imagem são representados por valores entre 0 e 255.

Fonte: homer.bmp



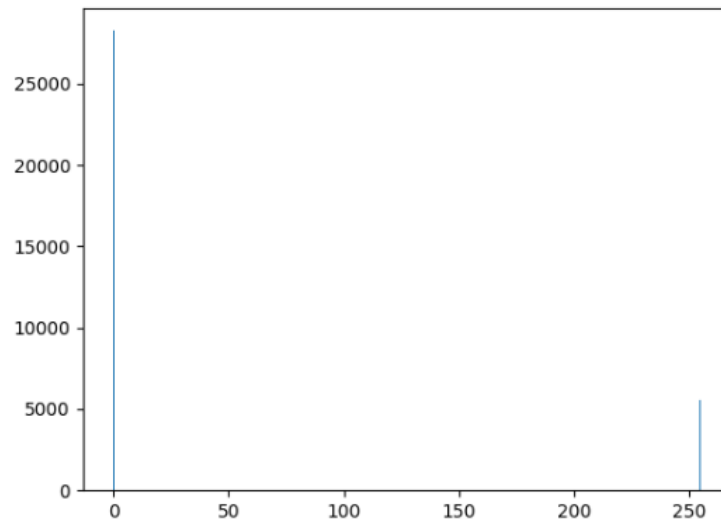
Como se trata de uma imagem a preto e branco com alguns tons de cinzento, tal como a imagem kid.bmp, irão aparecer valores entre 0 e 255 só que, como se trata de uma imagem mais pequena, o número de ocorrências destes valores será inferior aos do gráfico da imagem kid.bmp.

Fonte: guitarSolo.wav



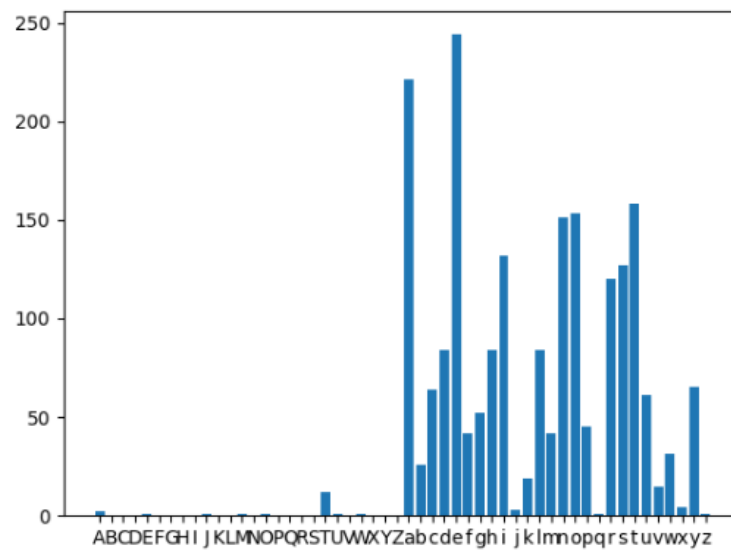
Trata-se de áudio que é representado por uma onda sonora, sendo os valores do gráfico as ocorrências de cada um dos valores dessa onda sonora.

Fonte: homerBin.bmp



Trata-se de uma imagem a preto e branco, mas desta vez apenas contém a cor preta e a cor branca (sem tons de cinza) pelo que os valores apenas são 0 e 255 (preto e branco).

Fonte: english.txt



Como english.txt se trata de um texto, irão ser listadas todas as letras do alfabeto, tanto maiúsculas como minúsculas. Notamos ainda no gráfico que o número de ocorrências de letras minúsculas é maior do que o das maiúsculas, como seria de esperar num texto.

É possível comprimir cada uma das fontes de forma não destrutiva, sendo a compressão máxima aquela que consegue alcançar um valor médio de bits por símbolo igual ao da entropia.

Nome do ficheiro	Entropia (bits/símbolo)	Taxa de compressão máxima (%)
kid.bmp	6.9541	13.0732
homer.bmp	3.4659	56.6767
homerBin.bmp	0.6448	91.9402
guitarSolo.wav	7.3292	8.3850
english.txt	4.2280	28.9918

Exercício 4

No exercício 4, é-nos pedido para, usando as funções de codificação de Huffman que são fornecidas, determinar o número médio de bits por símbolo para cada uma das fontes de informação. Através dos resultados obtidos, e como seria de esperar, observamos que o valor da entropia é inferior ao da média de bits por símbolo de todas as fontes.

Nome do ficheiro	Entropia (bits/símbolo)	Média de bits por símbolo	Variância (bits/símbolo)
kid.bmp	6.9541	6.9832	2.0994
homer.bmp	3.4659	3.5483	13.1968
homerBin.bmp	0.6448	1.1644	0.1374
guitarSolo.wav	7.3292	7.3502	4.2523
english.txt	4.2280	4.2523	1.1979

É possível reduzir a variância, e, para obter códigos de Huffman com variância mínima, devem-se colocar na lista os símbolos combinados utilizando a ordem mais elevada possível, isto é, devem-se agrupar os símbolos dois a dois o mais acima possível na árvore, de modo a criar uma árvore com menos profundidade, e códigos de comprimento mais uniforme. É útil, por exemplo, em casos de congestionamento de rede, em que se aplica um buffer que convém ter uma taxa de enchimento aproximadamente constante, e, por isso, é conveniente usar códigos de comprimento também aproximadamente constante, isto é, códigos de variância mínima.

Exercício 5

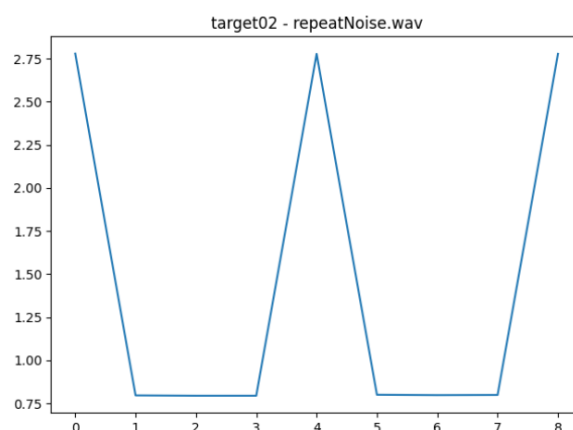
No exercício 5, é-nos pedido para determinar a distribuição estatística e o limite mínimo para o número médio de bits por agrupamento de símbolos (entropia), isto é, admitindo que cada símbolo é na verdade uma sequência de dois símbolos contíguos. Para isso, criámos a função 'agrupa', que dado uma tabela (fonte de informação), devolve outra tabela constituída por listas de dois elementos contíguos da passada como argumento.

Exercício 6

Alínea b)

```
Informação Mútua de "target01 - repeat.wav" = [7.3174 0.7743 0.7799 0.7739 7.3066 0.7752 0.7799 0.7746 7.3077]  
Informação Mútua de "target02 - repeatNoise.wav" = [2.7790 0.7964 0.7944 0.7945 2.7777 0.8002 0.7980 0.7991 2.7778]
```

Concluimos que os valores de informação mútua entre a query e o target representam o nível de semelhança entre estes, sendo que os valores mais altos correspondem aos momentos em que são mais semelhantes. Variam praticamente da mesma forma ao longo do tempo, embora de uma forma menos acentuada no ficheiro "target02 - repeatNoise.wav", como era expectável, visto que este consiste numa versão do ficheiro "target01 - repeat.wav" com ruído, logo é menos parecido com o original.



Alínea c)

Resultados:

- Song01: Informação Mútua = [0.6495 0.6335]
- Song02: Informação Mútua = [0.9182]
- Song03: Informação Mútua = [0.7756 0.7472]
- Song04: Informação Mútua = [1.1592 1.1646]

○ Song05:

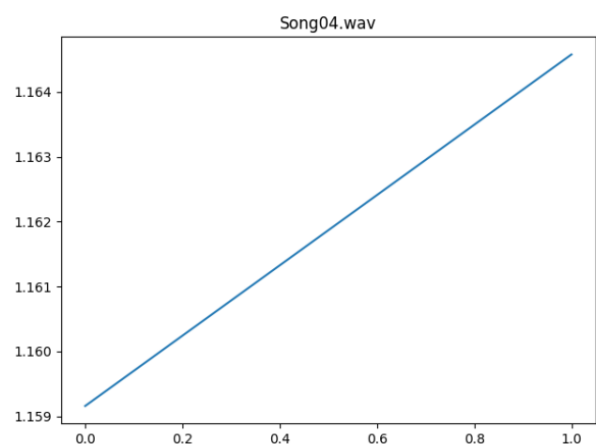
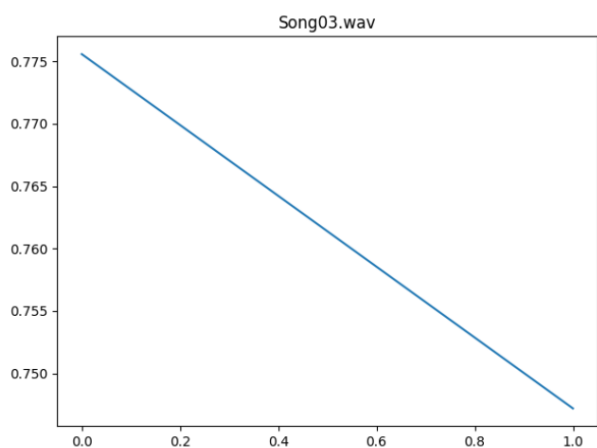
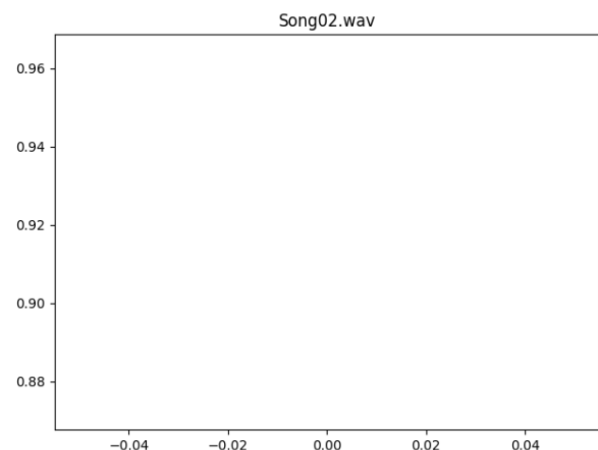
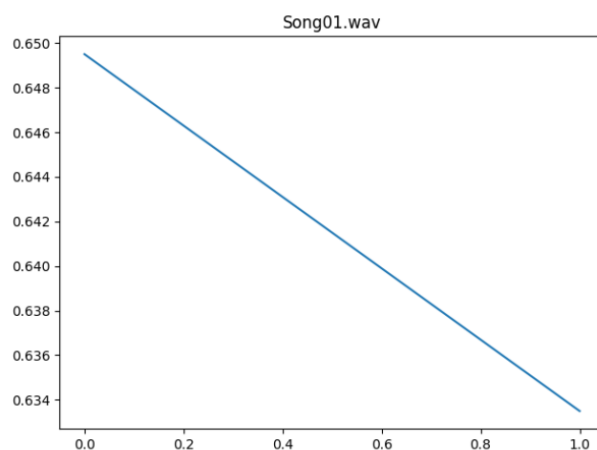
Informação Mútua = [4.0354 0.7757 0.7754 0.7750 0.7992 0.7860
0.7825 0.7874 0.7825 0.7774 0.7827 0.7936 0.8136 0.7993 0.7968 0.7805
0.7671 0.7540 0.7741 0.7758 0.7843 0.7888 0.7946 0.7967 0.7901 0.7937
0.8073 0.7765 0.7888]

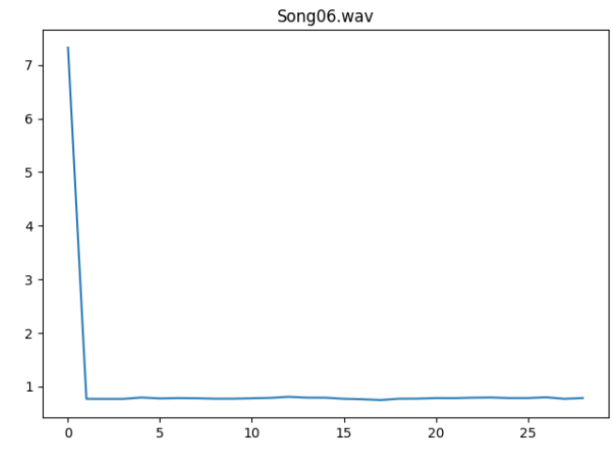
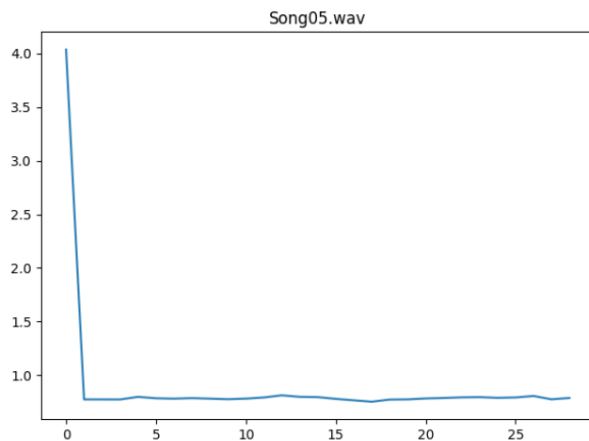
○ Song06:

Informação Mútua = [7.3226 0.7733 0.7714 0.7722 0.7975 0.7808
0.7868 0.7835 0.7757 0.7760 0.7834 0.7906 0.8111 0.7952 0.7939 0.7740
0.7654 0.7505 0.7752 0.7768 0.7870 0.7855 0.7943 0.7983 0.7870 0.7874
0.8019 0.7736 0.7878]

○ Song07:

Informação Mútua = [6.3139 0.4600 0.4641 0.4590 0.4785 0.4647
0.4671 0.4780 0.4643 0.4621 0.4620 0.4646 0.4838 0.4773 0.4802 0.4683
0.4529 0.4460 0.4661 0.4578 0.4700 0.4688 0.4769 0.4856 0.4747 0.4694
0.4864 0.4624 0.4893]





Máximo

Mínimo



Nome do ficheiro	Song06	Song07	Song05	Song04	Song02	Song03	Song01
Valor máximo da informação mútua (bits/símbolo)	7.3226	6.3139	4.0354	1.1646	0.9182	0.7756	0.6495

Nota: Apresentamos certos dados por escrito e não em imagem devido à dimensão dos mesmos.

Como referido na alínea anterior, os valores de informação mútua entre a query e o target representam o nível de semelhança entre estes, sendo que, mais uma vez, os valores mais altos correspondem a partes mais semelhantes. Assim, podemos concluir que o ficheiro “Song06.wav” é o que tem a parte mais parecida com o ficheiro original, devendo ser esta a música sugerida pelo identificador de música perante a original.