Entropia, Redundância e Informação. Mútua- Relatório

Teoria Da Informação

Universidade de Coimbra

FCTUC – Departamento de Engenharia Informática

Dinis Silva Costa Carvalho, 2018278118

João Pimentel Roque R. Teixeira, 2018278532

18 outubro 2019

Exercício 1:

Para a resolução deste problema, criámos uma rotina que devolve um histograma de ocorrências dos símbolos da fonte de informação que recebe. Esta rotina recebe também o alfabeto da fonte.

Exercício 2:

O limite mínimo teórico para o número médio de bits por símbolo corresponde ao valor da entropia da fonte de informação. Criámos, por isso, uma rotina que calcula a entropia de uma fonte, com base na seguinte fórmula:

$$H(A) = \sum_{i=1}^{n} P(a_i) i(a_i) = \sum_{i=1}^{n} P(a_i) log_2(P(a_i))$$

"A" corresponde ao alfabeto que a rotina recebe (A={a_1 , … , a_n }), e P(a_i) à probabilidade do acontecimento a_i.

Esta rotina apoia-se numa rotina auxiliar que calcula a probabilidade de a_i à medida que ciclo que representa o somatório vai sendo iterado.

A função recebe a fonte de informação e o alfabeto.

Exercício 3:

Tal como é pedido no enunciado, baseámo-nos nas rotinas que desenvolvemos em 1) e 2) para resolver este exercício. No entanto, estas rotinas não eram suficientes para tratar a informação da forma que necessitávamos. Criámos para isso a rotina "imagem.m", a rotina "som.m" e a rotina "texto.m" de forma a, dependendo do tipo de informação com que nos deparássemos, tivéssemos as ferramentas para lidar com ela, visto que os alfabetos são diferentes em cada caso.

Para a imagem, calculamos o alfabeto a partir da *BitDepth* da imagem fornecida, correspondendo o alfabeto a [0, 2^{NrBits}-1].

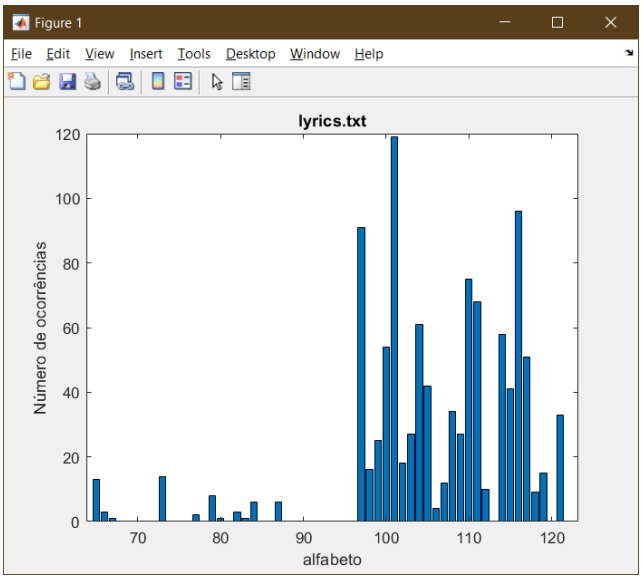
Para o som, analisámos a amostra de forma a obter o número de bits por sample, correspondendo o alfabeto a {-1, -1+$2^{NrBits-1}$, -1+2*$2^{NrBits-1}$,..., 1-$2^{NrBits-1}$}.

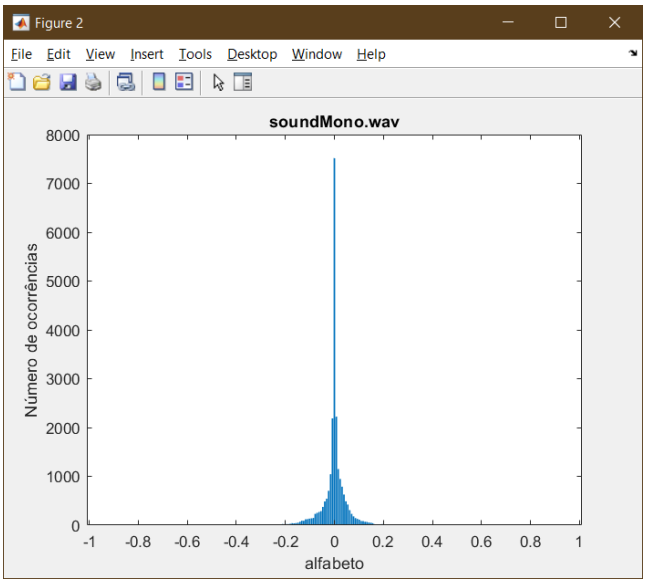
Quanto ao texto, criámos manualmente um alfabeto que contém as letras de a-z e de A-Z. O texto é lido a partir do ficheiro, e os seus carateres são contados. Para a construção do histograma, convertemos os carateres nos seus *double* correspondentes da tabela ASCII.

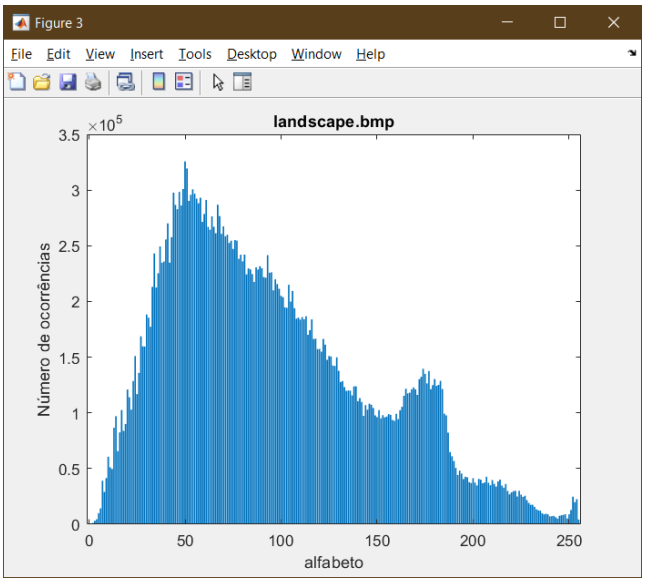
Utilizamos as funções *unique* e *reshape* de forma a ter os símbolos organizados e sem repetições, para o histograma ser criado corretamente.

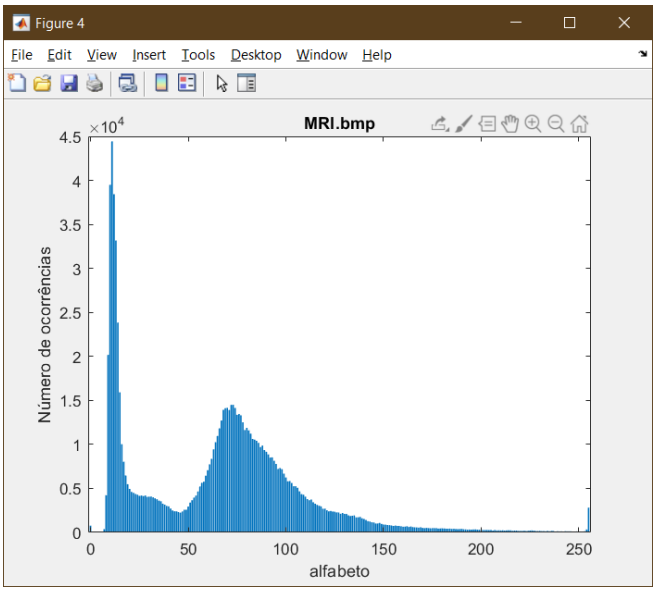
Podemos assim analisar os histogramas e os valores de entropia obtidos:

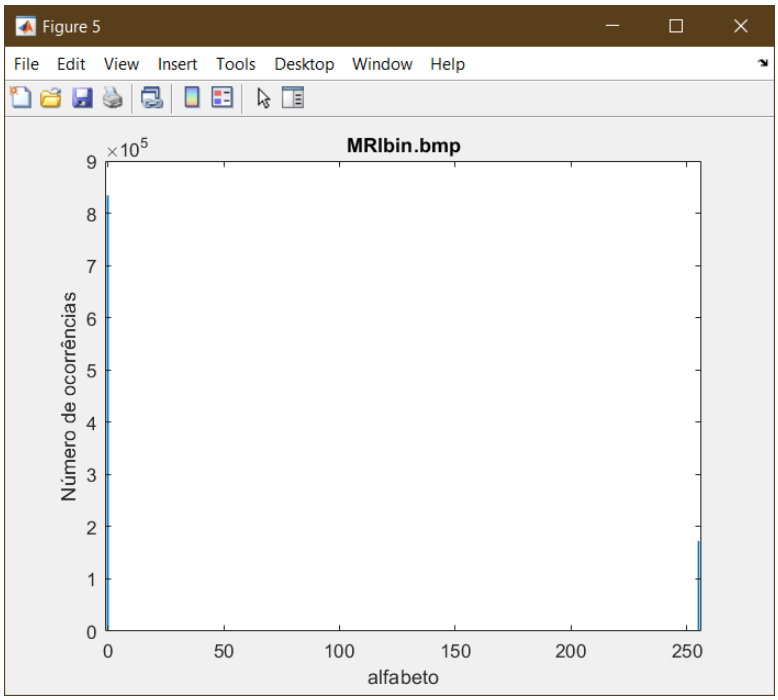
- landscape.bmp – 7.6069 bits/símbolo
- MRI.bmp – 6.8605 bits/símbolo
- MRIbin.bmp – 0.6611 bits/símbolo
- soundMono.wav – 4.0657 bits/símbolo
- lyrics.txt – 4.5409 bits/símbolo











Calculámos também a taxa de compressão de cada fonte com a seguinte fórmula:

$$Taxa\ de\ Compressão(X) = \frac{EntropiaMáxima(X) - Entropia(X)}{Entropia(X)} \times 100$$

Com base nestes valores, podemos tirar conclusões quanto à possibilidade de comprimir cada uma das fontes de forma não destrutiva:

- landscape.bmp – 4.9136%
- MRI.bmp – 14.2432%
- MRIbin.bmp – 91.7365%
- soundMono.wav – 49.2141%
- lyrics.txt – 20.3404%

Como em nenhum dos casos a taxa de compressão é nula, é possível comprimir os ficheiros de forma não destrutiva.

Exercício 4:

Para a resolução deste exercício utilizámos a função fornecida *hufflen.m*, e criámos a função *Huffman_entropia.m* que recebe o resultado da função *hufflen.m* e a matriz de probabilidades da fonte, obtida com a função *Freqocurr.m*, e devolve o valor do número médio de bits do código de Huffman de cada fonte, que apresentamos a seguir:

- landscape.bmp – 7.6544e-05 bits/simbolo
- MRI.bmp – 0.0075 bits/simbolo
- MRIbin.bmp – 0.8285 bits/simbolo
- soundMono.wav – 4.1107 bits/simbolo
- lyrics.txt – 4.4435 bits/simbolo

Os valores sublinhados (*landscape.bmp* e *MRI.bmp*) parecem-nos desadequados e consideramos que estarão errados. No entanto, após bastante escrutínio do código, não conseguimos identificar o erro que estará a causar estas anomalias.

Variância dos comprimentos dos códigos resultantes:

- landscape.bmp – 0.7516
- MRI.bmp – 2.1931
- MRIbin.bmp – 0
- soundMono.wav – 4.3335
- lyrics.txt – 1.0821

Exercício 5:

Neste exercício vamos novamente calcular a entropia e os histogramas destas fontes, mas desta vez procedemos ao agrupamento dos simbolos das fontes dois a dois, sendo que no caso de a fonte ter um número ímpar de elementos, retiramos o último símbolo. Cada símbolo irá ser a junção de dois símbolos, obtidos com a função *agrupamento.m*. O alfabeto destas novas fontes será o resultado da operação *unique* sobre as fontes obtidas no agrupamento. Vamos então calcular a entropia sobre as fontes alteradas:

- landscape.bmp – 6.2041 bits/símbolo
- MRI.bmp – 5.2269 bits/símbolo
- MRIbin.bmp – 0.4007 bits/símbolo
- soundMono.wav – 3.3110 bits/símbolo
- lyrics.txt – 3.5957 bits/símbolo

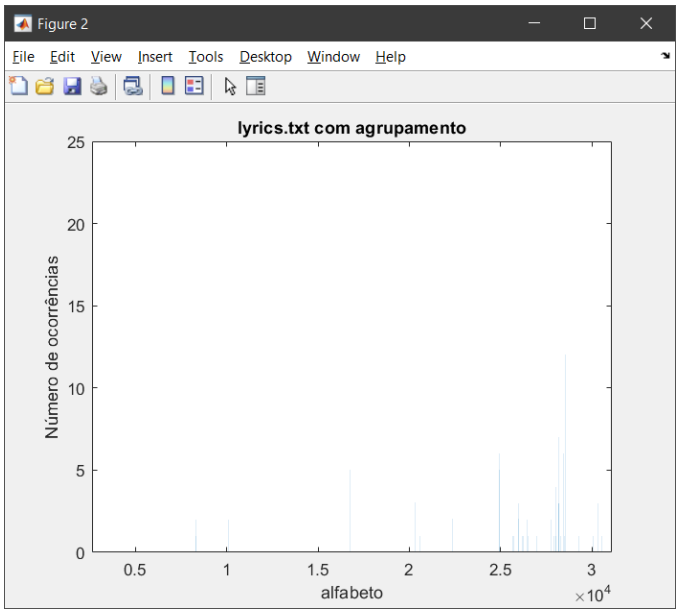


Figure 2
File Edit View Insert Tools Desktop Window Help
lyrics.txt com agrupamento
Número de ocorrências
alfabeto
×10^4

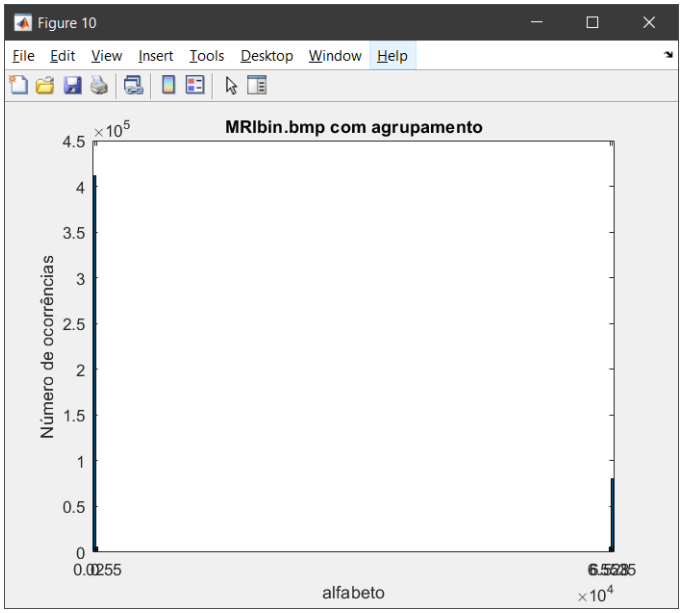


Figure 10
File Edit View Insert Tools Desktop Window Help
MRIbin.bmp com agrupamento
×10^5
Número de ocorrências
alfabeto
×10^4

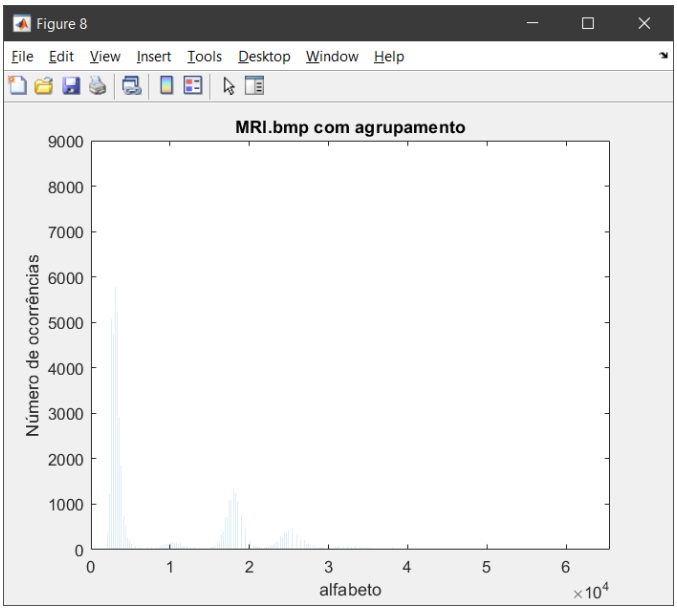


Figure 8
File Edit View Insert Tools Desktop Window Help
MRI.bmp com agrupamento
Número de ocorrências
alfabeto
×10^4

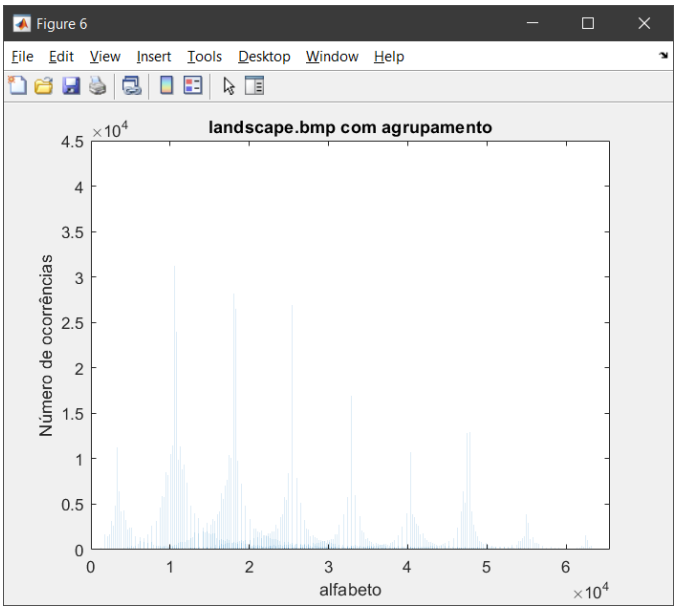


Figure 6
File Edit View Insert Tools Desktop Window Help
landscape.bmp com agrupamento
×10^4
Número de ocorrências
alfabeto
×10^4

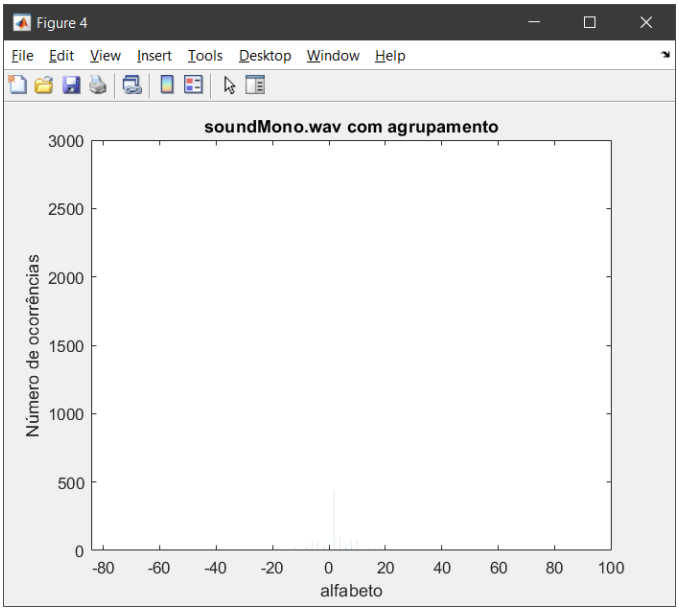


Figure 4
File Edit View Insert Tools Desktop Window Help
soundMono.wav com agrupamento
Número de ocorrências
alfabeto

Analisando os resultados, podemos concluir que a entropia diminui se agruparmos os símbolos dois a dois.

Exercício 6:

Criámos a função *infomutua.m*, que recebe como argumento o target, uma query, um alfabeto e o *step*. Esta função devolve uma matriz de valores da informação mútua de cada janela gerada.

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

É assim possivel calcular o valor da informação mútua entre a query (X) e o target (Y).

Recorrendo às funções *max* e *find*, conseguimos encontrar o valor máximo da informação mútua e as suas coordenadas na matriz de informações mútuas, que irá corresponder à melhor correspondência entre o target e a query.

Informação mútua máxima entre o *query.bmp* e as imagens fornecidas:

- 'target_original.bmp': 1.35
- 'target_noise.bmp': 1.1963
- 'target_lightning_contrast.bmp': 1.224
- 'target_inverted.bmp': 1.35
- 'target1.bmp': 1.7008
- 'target2.bmp': 1.2007
- 'target3.bmp': 1.1706
- 'target4.bmp': 1.1123

O *target_noise.bmp* e o *target_lightning_contrast.bmp* correspondem a alterações ao *target_original.bmp*, daí a menor informação mútua. O *target1.bmp* corresponde ao maior valor de informação mútua.