

# MBA em Ciência de Dados

## Técnicas Avançadas de Captura e Tratamento de Dados

### Módulo VII - Dados não estruturados: sinais e imagens

#### Exercícios

Moacir Antonelli Ponti

CeMEAI - ICMC/USP São Carlos

---

**Recomenda-se fortemente que os exercícios sejam feitos sem consultar as respostas antecipadamente.**

---

```
In [1]: # carregando as bibliotecas necessárias
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

#### Exercício 1)

Quando comparamos imagens e sinais e suas características, o que devemos considerar a priori?

- (a) Sinais possuem valores independente e identicamente distribuídos, enquanto Imagens possuem pixels organizados de forma espacial
- (b) Sinais possuem valores codificados em 16 bits, enquanto imagens possuem valores codificados em 8 bits
- (c) Sinais possuem valores com dependência sequencial, enquanto imagens não possuem padrão de dependência
- (d) Sinais possuem valores com dependência sequencial, enquanto Imagens possuem dependência espacial de seus valores

## Exercício 2)

Carregue os dados do arquivo `sinais2.csv` utilizando o comando:

```
signals = np.genfromtxt(arquivo, delimiter=',').astype(np.float32)
```

Esse array possui um sinal por linha `signals[i]`. Calcule a autocorrelação (usada para identificar padrões de repetição) de cada sinal (utilizando a função vista em aula). Plote os sinais e o valor absoluto de suas respectivas autocorrelações considerando deslocamentos (*lags*) de 1 até 50.

Calcule também o desvio padrão `np.std()` da autocorrelação de 1 até 50 de cada sinal.

Observe que um dos sinais possui desvio padrão maior relativo à autocorrelação e analise o padrão da autocorrelação exibida no gráfico. Considerando a posição do sinal no array (de 0 a 4), escolha a alternativa verdadeira:

- (a) A análise de autocorrelação mostra que os sinais 0 e 1 são sinais similares, sendo os demais bastante diferentes como evidenciado pelo desvio padrão da autocorrelação.
- (b) A análise de autocorrelação mostra que o sinal 3 tem desvio padrão superior e gráfico mais instável, indicando que esse possa ser um sinal com menos padrões de repetição, e portanto menor dependência temporal do que os demais.
- (c) A análise de autocorrelação indica que os sinais 0, 1, 2 e 3 possuem menor dependência temporal do que o sinal na posição 4, o qual é muito diferente dos demais sinais apresentando valores inferiores na análise.
- (d) O desvio padrão da autocorrelação mostra que o sinal da posição 4 tem valor inferior aos demais, indicando que esse possa ser um sinal mais ruidoso.

### Exercício 3)

Utilizando ainda os sinais carregados na questão anterior `sinais2.csv`, utilize a `np.fft.fft()` para obter a Transformada de Fourier dos sinais. Depois, considerando apenas frequências até 50, calcule quais são as 4 frequências de maior valor de magnitude (obtido pelo `np.abs()`). Aqui não queremos os valores da magnitude, mas a quais frequências (índices) elas se referem. Para complementar a análise, plote as magnitudes das transformadas até a frequência 50.

Analisando as frequências de maior magnitude temos as frequências que mais caracterizam o sinal. Considerando as 4 frequências computadas anteriormente, podemos dividir os sinais em categorias distintas. Nesse sentido, qual análise abaixo está correta?

- (a) O sinal 4 possui frequências inferiores quando comparado com os demais, indicando que o sinal 4 é provavelmente dependente sequencialmente, enquanto os demais são i.i.d.; assim podemos dividi-los em duas categorias: sinal 4 e sinais 0, 1, 2 e 3.
- (b) O sinal 3 possui frequências mais significativas 20 Hz ou superior, indicando que é um sinal com maior qualidade de aquisição, e assim podemos categorizar em: sinal 3, e sinais 0, 1, 2 e 4.
- (c) Todas as frequências estão abaixo de 50 Hz, sendo assim podemos dizer que os sinais são todos similares, sendo impossível dividi-los em categorias.
- (d) O sinal 3 possui frequências mais significativas 20 Hz ou superior, possuindo transições mais rápidas de valores do que os outros com frequências características menores do que 12Hz; e assim podemos categorizar em: sinal 3, e sinais 0, 1, 2 e 4.

### Exercício 4)

Considerando os mesmos sinais carregados, compute as características: entropia da energia (com 10 blocos), taxa de cruzamentos por zero, entropia espectral (com 10 blocos), formando um vetor com 3 características para cada sinal.

Após isso, compute a matriz de distâncias entre os 4 sinais considerando a distância L1, i.e., a soma dos valores absolutos das diferenças entre dois vetores  $A$  e  $B$ :

$$\sum_i |A_i - B_i|$$

Da matriz, que indica a dissimilaridade entre pares de sinais, aplique uma soma na direção do eixo 0 (axis=0) e depois arredonde para inteiro `np.round(, 0)`. Quais valores foram obtidos para cada sinal?

- (a) Sinais 0, 1, 2 e 4, soma 2; Sinal 3, soma 7.
- (b) Sinais 0 e 4, soma 3; Sinais 1 e 2, soma 2; Sinal 3, soma 7.
- (c) Sinais 0, 1, e 2, soma 2; Sinal 3, soma 7; Sinal 4, soma 3.
- (d) Sinais 0, 1, e 2, soma 1; Sinal 3, soma 5; Sinal 4, soma 2.

```

In [4]: def entropia_energia(sinal, n_blocos=10):
        '''Entropia da energia do sinal'''
        # energia total
        energia_sinal = np.sum(sinal ** 2)
        M = len(sinal)

        # calcula janelas dentro do sinal
        M_janelas = int(np.floor(M / n_blocos))
        # verifica se tamanho dos blocos
        # é multiplo do tamanho do sinal
        if M != M_janelas * n_blocos:
            sinal = sinal[0:M_janelas * n_blocos]

        # monta matriz [M_janelas x n_blocos]
        janelas = sinal.reshape(M_janelas, n_blocos, order='F')
        janelas = janelas.copy()

        # Computa energias de cada janela (normalizada pela do
        # sinal)
        e_janelas = np.sum(janelas ** 2, axis=0) / (energia_sinal + 0.0001)
        #print(e_janelas)

        # Computa entropia entre energias das janelas
        entropia = -np.sum(e_janelas * np.log2(e_janelas + 0.0001))
        return entropia

def taxa_cruzamentos_por_zero(sinal):
    '''Cruzamentos por zero em um intervalo de tempo '''
    M = len(sinal)
    cont_zero = np.sum(np.abs(np.diff(np.sign(sinal)))) / 2
    return np.float64(cont_zero) / np.float64(M - 1.0)

def entropia_espectral(sinal, n_blocos=16):
    """Computes the spectral entropy"""

    fft_abs = np.abs(np.fft.fft(sinal))

    entropia_esp = entropia_energia(fft_abs, n_blocos=n_blocos)

    return entropia_esp

```

## Exercício 5)

Carregue as seguintes imagens da base de dados flickr\_map\_training:

```
img1 = imageio.imread("dados/flickr_map_training/107.jpg")
img2 = imageio.imread("dados/flickr_map_training/101.jpg")
img3 = imageio.imread("dados/flickr_map_training/112.jpg")
img4 = imageio.imread("dados/flickr_map_training/303.jpg")
img5 = imageio.imread("dados/flickr_map_training/400.jpg")
```

Implemente um descritor de cor que computa um histograma utilizando a composição dos canais RGB em um único canal utilizando a seguinte operação, sendo R, G e B as matrizes relativas a cada canal de cor:

$$I = R \cdot 0.3 + G \cdot 0.59 + B \cdot 0.11$$

Permita definir o número de bins do histograma por meio da sua função e, antes de retornar, normalize o histograma dividindo pela soma.

Depois, calcule a distância entre img1 carregada e as outras imagens (2, 3, 4, 5) utilizando: 16 bins e 4 bins. Qual foram as duas imagens mais similares, da mais próxima para a mais distante, nos dois casos?

- (a) 16 bins: img2, img4 ; 4 bins: img2, img3
- (a) 16 bins: img2, img3 ; 4 bins: img4, img3
- (b) 16 bins: img2, img3 ; 4 bins: img2, img4
- (d) 16 bins: img4, img2 ; 4 bins: img4, img3

## Exercício 6)

Vamos repetir o procedimento da questão anterior, agora utilizando o descritor de texturas LBP visto em aula. Utilizaremos uma função que também realiza uma normalização dos valores máximos das imagens, bem como permite definir o raio, número de pontos e quantidade de bins para esse descritor, conforme abaixo.

Calcule a distância L1 entre img1 carregada e as outras imagens utilizando o descritor LBP com os seguintes parâmetros:

- número de pontos = 14
- raio = 2
- bins = 16

Quais foram as três imagens mais similares, da mais próxima para a mais distante?

- (a) img3, img2, img5
- (b) img2, img3, img4
- (c) img3, img5, img2
- (d) img5, img3, img2

```
In [1]: def lbp_features(img, points=8, radius=1, n_bins=10):
        # LBP opera em imagens de um só canal, aqui vamos converter
        # RGB para escala de cinza usando o método Luminance
        img = np.array(img, dtype=np.float64, copy=False)
        if (len(img.shape) > 2):
            img = img[:, :, 0]*0.3 + img[:, :, 1]*0.59 + img[:, :, 2]
            *0.11

        # normaliza a imagem para ter máximo = 255
        if (np.max(img) > 0):
            img = ((img/np.max(img))*255).astype(np.uint8)

        # aqui definimos o numero de pontos e o raio, padrao =
        8, 1
        lbp = feature.local_binary_pattern(img.astype(np.uint
        8), points, radius, method="uniform")

        # lbp retorna um matriz com os códigos, então devemos e
        xtraír o histograma
        (hist, _) = np.histogram(lbp.ravel(), bins=np.arange(0,
        n_bins+1), range=(0, n_bins))

        # normaliza o histograma
        hist = hist.astype("float")
        hist /= (hist.sum() + 1e-6)
        # return the histogram of Local Binary Patterns

        return hist
```

### Exercício 7)

No método Bag-of-Features quais dos parâmetros pertencem ao *framework* influenciam mais drasticamente a performance do método no caso de uso em imagens?

- (a) O tamanho do dicionário, a quantidade de cores nas imagens, a quantidade de classes do problema
- (b) O tamanho do dicionário, o descritor base, o método utilizado para aprender o dicionário
- (c) O descritor base e o número de componentes principais utilizados
- (d) O tamanho do patch extraído da imagem, que deve ser compatível com a resolução das imagens

### Exercício 8)

Execute o método Bag-of-Features estudado em aula nas imagens da pasta `flickr_map_training`, agora com os seguintes parâmetros:

- tamanho do patch = (13, 13)
- número de patches = 1000
- descritor base = PCA com 10 componentes principais
- tamanho do dicionário = 50

Utilize imagens de consulta `flower.jpg` e `football.jpg` e recupere as 12 imagens mais similares utilizando o modelo BoF aprendido. Qual a proporção de imagens retornadas da mesma categoria da consulta?

- (a) flower = 3/12, football = 3/12
- (b) flower = 5/12, football = 2/12
- (c) flower = 6/12, football = 0/12
- (d) flower = 9/12, football = 1/12

## Exercício 9)

Execute novamente o método Bag-of-Features conforme questão anterior, porém agora utilizaremos um descritor base diferente.

- tamanho do patch = (13, 13)
- número de patches = 1000
- tamanho do dicionário = 50
- descritor base = LBP com raio 2, 16 pontos e 10 bins

Vamos usar a versão da função LBP que permite usar como parâmetros o número de pontos e raio.

Utilize imagens de consulta `flower.jpg` e `football.jpg` e recupere as 12 imagens mais similares utilizando o modelo BoF aprendido. Qual a proporção de imagens da mesma categoria da consulta?

- (a) flower = 5/12, football = 3/12
- (b) flower = 6/12, football = 0/12
- (c) flower = 6/12, football = 2/12
- (d) flower = 6/12, football = 6/12



## Exercício 10)

Execute o método Bag-of-Features para aprender features nas imagens da pasta `flickr_map_training` conforme código fornecido em aula, com os seguintes parâmetros:

- tamanho do patch = (11, 11)
- número de patches = 300
- descritor base = PCA com 16 componentes
- `random_state` = 1
- para o KMeans use `random_state=1` e `n_init=3`

Vamos investigar a influência do tamanho do dicionário no modelo gerado com os seguintes valores: 8, 16, 32, 64, 128, 256 e 512. Utilize a imagem de teste

`flickr_map_test\flower.jpg` para recuperar as 16 imagens mais similares no conjunto de treinamento (sabendo que há 10 imagens dessa categoria no conjunto de treinamento). Calcule a revocação, ou seja, a razão entre o total de imagens de flores retornadas na busca das 16 mais similares e o número total de imagens de flores que deveriam ter sido retornadas (10).

DICA: as imagens de flores tem nome iniciando com o número '6'.

Quais tamanhos de dicionário resultam em maior revocação?

- (a) 256 e 512
- (b) 64 e 128
- (c) 32, 64, 128 e 256
- (d) 64, 128 e 256