

IMPORTANT

Si una pregunta no la contesteu, poseu un 0. Això us donarà un número de 13 xifres com a "Resultat respostes". Apunteu-vos el "CODI EXAMEN" i el "Resultat respostes". Ompliu l'enquesta que trobareu a cerbero "Test projecte 2". Aquest codi numèric ens permetrà donar-vos la nota dilluns.

Nota 1: per a executar el codi necessiteu el python 2.7.x. En els ordinadors de les aules podeu accedir-hi usant Spyder, Idle, o amb command line. En aquest últim cas, quan estigueu en el prompt de sistema heu d'introduir `set path=c:\anaconda;%path%`

Nota 2: podem inicialitzar una matriu amb uns nombres donats amb el següent codi:

```
X = np.array([[239, 44, 35], [179, 267, 254], ...])
```

Nota 3: si es vol obtenir les dades de la inicialització, o veure quin és el resultat que s'obté d'aplicar únicament un sol pas de l'algorisme de kmeans podeu cridar a

```
k_m = km.KMeans(X,K,options)
```

```
k_m._init_centroids() o k_m._get_centroids() o ....
```

i accedir a les dades de `k_m` o usar `print` en el lloc que considereu oportú.

Nota 4: En les preguntes que opereu sobre imatges NO re-escalar les imatges

(eliminar `im = rescale(im, 0.25, preserve_range=True)` del fitxer `TestKmeans.py`).

Les preguntes 1, 2, i 3 fan referència al següent conjunt de punts i centroides:

Punt	Valors
1	<239, 44, 35>
2	<179, 267, 254>
3	<175, 158, 1>
4	<204, 21, 32>
5	<284, 205, 239>
6	<265, 58, 279>

Centroide	Valors
1	<211, 116, 22>
2	<202, 13, 50>

1. Quina és la distància euclideana del punt 6 al tots els centroides?

1) 72329.00, 58435.00

2) 3501.00, 24155.00

3) 59.17, 155.42

4) 268.94, 241.73

Resposta: 4

```
X = np.array([[265,58,279]])
```

```
C = np.array([[211.0, 116.0, 22.0], [202.0, 13.0, 50.0]])
```

```
D = km.distance(X,C)
```

```
print(D)
```

$$\sqrt{(211-265)^2 + (116-58)^2 + (22-279)^2} = 268.94$$
$$\sqrt{(202-265)^2 + (13-58)^2 + (50-279)^2} = 241.73$$

2. A quin centroide assignarem cada un dels punts?

1) 1, 2, 2, 1, 2, 1

2) 2, 1, 1, 2, 1, 2

3) 2, 1, 1, 1, 1, 2

4) 2, 1, 1, 2, 2, 1

Resposta: NS/NC

```
print("\nCluster Points:")
```

```
k_m._cluster_points()
```

```
D_1=k_m.clusters[:100]
```

```
print(D_1)
```

A mi em dona una cosa molt diferent així que NPI...

```
Cluster Points:
[0 1 0 0 1 1]
```


3. Si iterem l'algorisme KMeans 1 cop a partir de les dades que tenim, quins centroides obtindrem?

Resposta: 4

- 1) <231.50, 236.00, 246.50>, <220.75, 70.25, 86.75>
- 2) <212.67, 210.00, 164.67>, <236.00, 41.00, 115.33>
- 3) <211.00, 116.00, 22.00>, <202.00, 13.00, 50.00>
- 4) <206.00, 74.33, 22.67>, <242.67, 176.67, 257.33>

```
X = np.array([dades array])
```

```
k_m = km.KMeans(X, 2, options)
```

```
k_m.centroids = np.array([dades centroides])
```

```
k_m._get_centroids()
```

```
print(k_m.centroids)
```

4. Si l'espai inicial és *Color Naming* i volem 9 centroides a partir de 250 punts, quina serà la mida de la matriu de centroides?

1) 9×11

Resposta: 1

2) 9×3

Segons l'enunciat tenim 9 centroides i l'espai inicial és Color Naming, que conté 11 colors.

3) 250×1

4) 250×11

5. Si la imatge és de 150×350 píxels i tenim 5 centroides, quina és la mida del vector que indica la pertinença d'un punt a un cluster?

1) 52500×3

Resposta: 2

2) 52500×1

Representa que $150 \times 350 = 52500$

3) $(150 \times 350) \times 3$

Llavors el vector és $P \times 1$ on $P = N \times M$

4) $(150 \times 350) \times 11$

S'inicialitza a `_init_X`, funció que no retorna res, però inicialitza la propietat `self.X` com un vector de mida $P \times D$, on $P = N \times M$. (D representa la dimensió)

6. Donada la imatge 0065.jpg quins centroides obtindrem si el mètode d'inicialització és 'first' amb $K=3$, en l'espai RGB? Nota: l'ordre en que apareixen els centroides no és rellevant.

1) <237.00, 216.00, 187.00>, <236.00, 215.00, 186.00>, <242.00, 221.00, 192.00>

2) <186.56, 154.58, 119.01>, <115.31, 106.77, 96.66>, <227.38, 199.24, 163.44>, <58.64, 55.52, 51.49>

3) <120.45, 110.68, 99.48>, <60.10, 56.91, 52.79>, <199.96, 168.40, 131.91>

4) <88.03, 82.34, 75.34>, <187.53, 155.23, 118.77>, <226.75, 201.11, 169.53>

```
Original:
[[120.56056537 110.74204947 99.4429682 ]
 [ 60.13886046  56.95206414  52.75997953]
 [199.94649784 168.43211593 131.98287345]]
```

```
im = io.imread('Images/0065.png')
options = {'verbose': False, 'km_init': 'first'}
k_m = km.KMeans(im, 3, options)
k_m.run()
print("Original:")
print(k_m.centroids)
```

DebatS, %Dubto?

7. Considerant $K=3$ i si afegim una nova opció en el mètode de inicialització que anomenarem 'custom' que executa el següent codi:

```
self.centroids = np.zeros((self.K, self.X.shape[1]))
for k in range(self.K): self.centroids[k,:] = k*255/(self.K-1)
```

quins són els centroides resultat de la inicialització per a la imatge 0065.jpg?

1) <34, 33, 31>, <53, 51, 52>, <223, 203, 179>

Resposta: 4

2) <1, 0, 0>, <128, 126, 117>, <255, 252, 234>

print("Exercici 7.")

3) <237, 216, 187>, <236, 215, 186>, <242, 221, 192>

exe_7 = km.KMeans(im, 3, options)

4) <0, 0, 0>, <127, 127, 127>, <255, 255, 255>

print(exe_7.centroids)

```
def _custom(self, K):
    self.centroids = np.zeros((self.K, self.X.shape[1]))
    for k in range(self.K):
        self.centroids[k,:] = k*255/(self.K-1)
```

```
Exercici 7.
Centroides Exe 7:
[[ 0.  0.  0.]
 [127.5 127.5 127.5]
 [255. 255. 255.]]
```

8. Considerant que la transformació d'un valor RGB a CIELab es fa usant el codi

```
from skimage import color
im = color.rgb2lab(im)
```

quin és el valor del píxel de la fila 41, columna 31 de la imatge 0065.jpg en format CIELab?

1) <88.97, 4.19, 11.96>

Resposta: 2

2) <72.76, 4.49, 12.51>

print("\nExercici 8.")

3) <196.00, 175.00, 156.00>

im = color.rgb2lab(im)

4) <241.00, 220.00, 201.00>

print(im[41][31])

Exercici 8.

[72.75814495 4.49006339 12.50778103]

9. Considerant que la transformació d'un valor RGB a *Color Naming* es fa usant el codi

```
import ColorNaming as cn
im = cn.ImColorNamingTSELabDescriptor(im)
```

quin és el valor del píxel de la fila 17, columna 65 de la imatge 0065 . jpg en format del *Color Naming*?

- 1) <0.00, 0.02, 0.00, 0.93, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00>
- 2) <0.00, 0.69, 0.00, 0.31, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00>
- 3) <241.00, 217.00, 191.00>
- 4) <254.00, 230.00, 206.00>

Resposta: 2 ¿Dubto?

```
print("\nExercici 9.")
im2 = cn.ImColorNamingTSELabDescriptor(im)
print(im2[17][65])
```

```
Exercici 9.
[0.00000000e+00 6.92134794e-01 0.00000000e+00 3.07691300e-01
 1.47123474e-04 7.68706534e-09 3.85440966e-13 2.67739828e-05
 0.00000000e+00 0.00000000e+00 0.00000000e+00]
```

Els numeros no son exactament els mateixos però son els més similars (Cal mirar l'array, amb el print no es veu exacte)

10. Si sabem que l'ordre de les dimensions de l'espai de *Color Naming* és: [Red, Orange, Brown, Yellow, Green, Blue, Purple, Pink, Black, Grey, White], i tenint en compte que el paràmetre `single_thr=0.7`, com etiquetariem el punt

<0.00, 0.85, 0.00, 0.14, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00>

- 1) GreenOrange
- 2) Orange
- 3) OrangeYellow
- 4) Yellow

Resposta: 2

Single_thr és el llindar del color, en aquest cas un 70% Podem veure que el 0.85 pertany a Orange i com que és més d'un 70% només orange. Cap altre té un 70% de color així que no pot ser OrangeYellow ja que només té un 0.14

11. Si sabem que l'ordre de les dimensions de l'espai de *Color Naming* és: [Red, Orange, Brown, Yellow, Green, Blue, Purple, Pink, Black, Grey, White], i tenint en compte que el paràmetre `single_thr=0.7`, com etiquetariem el punt

<0.00, 0.35, 0.00, 0.65, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00>

- 1) GreenYellow
- 2) Yellow
- 3) Orange
- 4) OrangeYellow

Resposta: 4

En aquest cas cap color té un marge superior al 70% per tant no és un color simple sinó compost. Veiem doncs que és OrangeYellow (0.35+0.65). Explicació detallada a pàgina 48 del PDF "PROJECTE2-Presentacio_2018_19.pdf"

12. Una de les mesures de similitud que podem calcular entre l'etiquetatge automàtic d'una imatge (*I*) i l'etiquetatge del Grountruth (*G*) és: $\frac{|I \cap G|}{|I|}$ on *|A|* indica el número d'elements del conjunt *A*.

Si el groundtruth de dues de les nostres imatges és:

Groundtruth 1:{ Black, Blue, Brown, BrownWhite, Orange, Yellow}
Groundtruth 2:{ Black, BlackGreen, Blue, Green, Grey, OrangeYellow, White, Yellow}

i l'etiquetatge automàtic ens ha donat respectivament:

Imatge 1:{ Yellow, Brown, Orange, BrownWhite, GreyWhite}
Imatge 2:{ Yellow, BrownYellow, GreyYellow, BlackOrange, BlackGrey, Black, GreenWhite, Blue}

Imatge 1	Groundtruth	Imatge2	Groundtruth
-	Black	Black	Black
-	Blue	BlackGrey	-
Brown	Brown	-	BlackGreen
Brown White	Brown White	BlackOrange	-
Grey White	-	Blue	Blue
Orange	Orange	BrownYellow	-
Yellow	Yellow	-	Green
		GreenWhite	-
S=4/5		-	Grey
		GreyYellow	-
		-	OrangeYellow
		-	White
		Yellow	Yellow
		S=3/8	
			S ₂ =0,375

S₁=0,8

S₂=0,375

quin són els valors de similitud entre etiquetatge automàtic i groundtruth per a les dues imatges?

- 1) 0.67, 0.38
- 2) 1.20, 1.00

Resposta:
Hauria de ser segons la formula (0.8, 0.375)

13. Considerant el següent pseudo-codi de la classe KMeans que ens ha de retornar un valor que ens mesuri com de bé s'ha adaptat el KMeans amb una determinada K a les dades que li hem entrat:

```
def fitting(self):
    if self.options["fitting"]=="Fisher":
        # calcular IntraClass()/InterClass()
        ....
    elif self.options["fitting"]=="Silhouette":
        # calcular la mitjana de (D1(x)-D2(x))/max(D1(x)-D2(x))
        # per a cada punt x
        ....
    return valor_calculat
```

Per a trobar la millor K necessitem incloure codi dins la funció processImage(). Aquest codi harà de processar diferent les dades provinents de la funció fitting(), segons la mesura que ens retorni:

```
def processImage(im, options):
    ...
    if options['K']<2:
        fit = []
        for k in range(2,K_maxima):
            kmeans = km.KMeans(im, k, options)
            fit.append( kmeans.fitting() )

    if options['fitting']=='Fisher':
        # Codi 1
        ...
    if options['fitting']=='Silhouette':
        # Codi 2
        ...
    ...
```

Quina de les següents afirmacions és certa? **RESPOSTA: NPI**

- 1) El "codi 1" haurà de calcular per quins valors de k el quocient entre la IntraClass i la InterClass es comença a fer gran.
- 2) La mètrica silhouette inclou en D1 i D2 informació de la distància entre les classes i la distància dins la classe per a cada punt de les dades
- 3) El càlcul de la distància InterClass es pot estimar calculant la distància entre els centroides representant les dues classes.
- 4) El "codi 2" haurà de calcular per quina k el valor de la mesura silhouette es té un punt d'inflexió.