TD 6

October 7, 2024

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering, SpectralClustering
```

Exercise 1 :

1 Load Wine dataset and save samples features in a matrix X and classes in a vector y.

2 Code a function to compute $n \times n$ matrix of Euclidian distances between n samples.

3 Perform the four clustering algorithms on X using *sklearn.cluster* functions.

4 Plot clustering results in all 2-d projections of X.

We denote $\ell(y_1, y_2)$ the classification loss function that equals to 1 if $y_1 = y_2$ and 0 otherwise.

5 Code this loss function and assess results of the four algorithms using true labels y.

6 Apply PCA on X and apply the different clustering methods on the first two principal components.

7 Plot the results on the first two principal components, compute their classification error using the loss function and compare to previous results.

Exercise 2 :

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as pltimg

from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering, SpectralClustering
```

1 Load the two provided images and plot them.

We want to flatten these images but keeping RGB and alpha values as features :

```
X = img.reshape((img.shape[0]*img.shape[1],-1))
```

2 Apply k-means algorithm on obtained matrices with 3, 5 and 10 clusters.

3 Replace, in a new matrix, each pixel by the cluster center value of its associated cluster and plot the results.

We propose now to add a notion of geometric distance and not only color distance. For that, we simply create a new matrix with labels of the first clustering as the first feature, and containing coordinates as other features.

```
y =  kmeans.predict(X)
XX = np.zeros((img.shape[0]*img.shape[1],3))

Cte = 10
XX = np.zeros((y.size,y.size))
a,b = np.unravel_index(np.arange(y.size),(img.shape[0],img.shape[1]))

for k in range(y.size):
    XX[k,:] = y[k],a[k]/Cte,b[k]/Cte
```

4 Suggest another way to take also into account pixel coordinate distances.

5 Apply DBSCAN and Spectral clustering on the new XX data and observe the results.