Practical work 08 – 7th of November 2023
Clustering algorithms

Summary for the organisation :

- Submit the solutions of the practical work before the date specified in Moodle.
- **Rule 1**. Submit an archive (***.zip** !) with your Python notebooks (one per exercise), including datasets and all necessary files.
- **Rule 2**. The archive file name must contain the number of the practical work, followed by the family names of the team members by alphabetical order, for example `02_dupont_muller_smith.zip`. Put also the name of the team members in the body of the notebook (in first cell). Only one submission per team.
- **Rule 3**. We give a **fail** for submissions that do not compile (missing files are a common source of errors...). So, make sure that your whole notebooks give the expected solutions by clearing all cells and running them all before submitting.

Context

The goal of this practical work is the implement by yourself the k-means algorithm and to experiment with the different parameters of this algorithm.

Exercice 1 Getting the data

a) Load the two given datasets :

```
data1, label1 = pickle.load(open("dataset_1.pkl", "rb"), encoding ="latin1")
data2, label2 = pickle.load(open("dataset_2.pkl", "rb"), encoding ="latin1")
```

b) Visualize the data using various color for each unique labels like in Figure 1 :

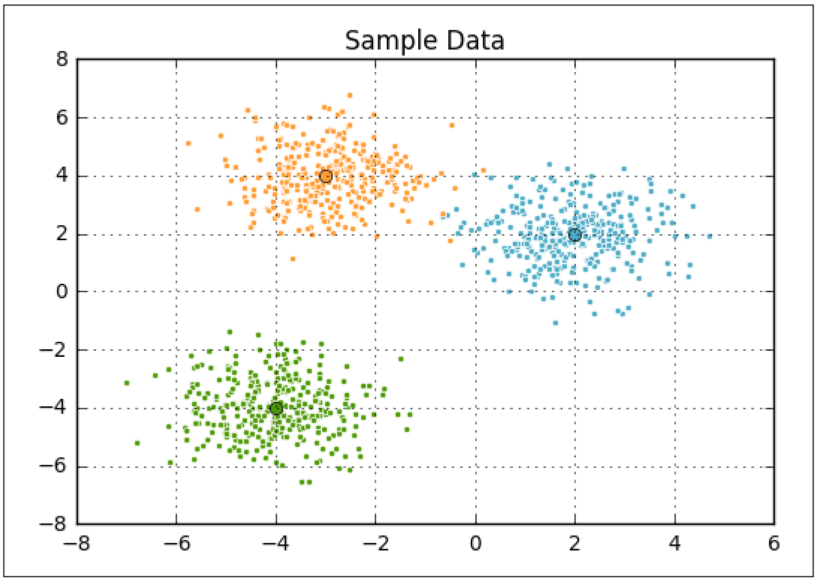

FIGURE 1 – Data visualization

Exercice 2 The k-means algorithm

Using numpy, implements the k-means algorithm as follow :

a) Initialise k centroids $\mu_1, \mu_2, \dots, \mu_K$.

b) Until convergence :

   i) Find the closest centroid for each training point

   ii) Reevaluate the centroids

c) Return the k centroids.

We also ask you to define and implement strategies for the :

- Initialisation of the centroids.
- Convergence criteria.

Exercice 3 Evaluate your model

At this point, your k-means algorithm is working :

- Visualize your convergence criteria over the epochs [1] using the dataset 1.
- Visualize the output of your k-means on the dataset 1.
- Do you experience sensitivity to the initial values of the centroids ? Is your strategy for initialization working well in most cases ?
- Document your convergence criteria. Could you think about other convergence criteria ?
- Visualize your convergence criteria over the epochs using the dataset 2.
- Visualize the output of your k-means on the dataset 2 and comment your results.

[1] One epoch is a complete visit of the training set.

Exercice 4 Optional : compare your implementation with the one of scikit-learn

Visit the page of `sci-kit learn` related to the K-Means algorithm and analyse the API : https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. Realise some experiments with this implementation and compare the results with the ones of your own implementation. What are your observations ?

Exercice 5 Review questions

a) Re-explain in your own words the steps of the K-Means algorithm.

b) Are we guaranteed to observe a decreasing distortion J from one epoch to the other in the K-Means ?

c) For two different initial values of the centroids in the K-Means, can we get different end values of the distortion J ? Argument your answer.

d) Can the K-Means be used as a compression algorithm ? Compute the compression ratio for a setting with 256 centroids and an input space at two dimensions (x_1, x_2) encoded in float32.

e) What is the use of the elbow method ? Explain it in your own words.

f) Give an example where we would know in advance the number of clusters we want to discover with a clustering algorithm.

g) It is possible to compute the distortion J_k for a given centroid k. If we observe that the distortion J_k for centroid k is really bigger than the other distortions and that the number of points N_k associated to this centroid is also bigger than for the other centroids, what can we say about the dataset and cluster k of points ? Could you suggest a strategy to make things better ?