INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Introduction to Machine Learning — 2024/2025
Unsupervised Learning

> These exercises should be solved using Python notebooks (Jupyter) due to the ability to generate a report integrated with the code. It is assumed you are proficient with programming. All answers must be justified and the results discussed and compared to the appropriate baselines.
> Max score of the assignment is 2 points. Extra exercises (if existing) help achieving the max score by complementing errors or mistakes.
> **Deadline:** end of class in week October, 14^{th}-18^{th}, 2024

This assignment will demonstrate how a learning algorithm can distinguish between two distributions of points generated with different parameters, using no information on the target values.

Exercise 1

Generate 2D points using a multivariate Gaussian distribution

1. Use the code in Fig. 1 to generate two sets, each with 500 points (reduce this number if necessary to obtain better visualizations or faster training runs),
2. Each dataset should have different centers, and sets should have a small overlap.
3. Add a column and fill it with 1 (one) for the first dataset and 2 (two) on the second, so that you can keep track of which distribution generated each point.
4. Join and shuffle the dataset.
5. The plot of the first two columns should be similar to the one presented in Fig. 2.
6. Write the dataset to a file.

```
import matplotlib.pyplot as plt
import numpy as np
import random

mean = [3, 3]
cov = [[1, 0], [0, 1]]
a = np.random.multivariate_normal(mean, cov, 500).T

mean = [-3, -3]
cov = [[2, 0], [0, 5]]
b = np.random.multivariate_normal(mean, cov, 500).T

c = np.concatenate((a, b), axis = 1)
c = c.T
np.random.shuffle(c)
c = c.T

x = c[0]
y = c[1]

plt.plot(x, y, 'x')
plt.axis('equal')
plt.show()
```

Figure 1: Code for generating two multivariate normal distributions

Implement a simple version of K-Means

1. Start by choosing two random points in the dataset r_1 and r_2 and apply the following adaptation rule:

   for all $x \in$ the dataset **do**
   if x is closer to r_1 than to r_2 **then**
   $r_1 \leftarrow (1-\alpha) \times r_1 + \alpha \times x$
   else if x is closer to r_2 than to r_1 **then**
   $r_2 \leftarrow (1-\alpha) \times r_2 + \alpha \times x$
   end if
   end for

2. Repeat for 10 times a passage through all the elements of the dataset (i.e. 10 epochs) with $\alpha = 10E - 5$ and save:

   (i) the consecutive values of r_1 and r_2 for the first passage;

   (ii) the values of r_1 and r_2 at the end of each passage.

3. Plot (i) and (ii) upon the dataset plot in different graphs. Change the value of α and the number of epochs to see the evolution of the representatives clearly. What do you conclude about the evolution of the two points in the different situations? Is there any

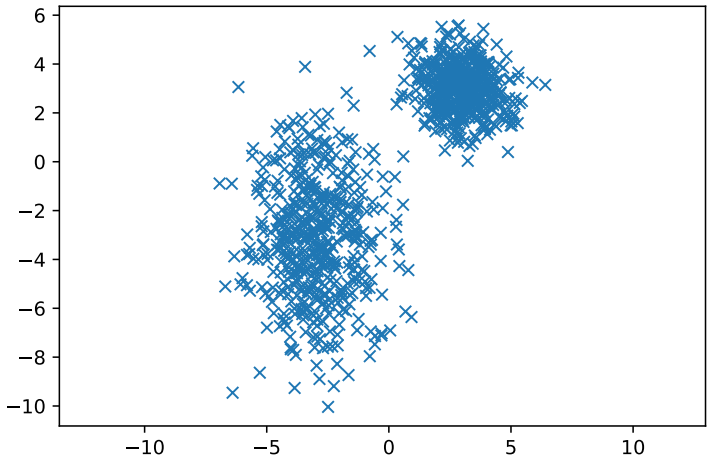

Figure 2: 2D points generated from two multivariate Gaussian distributions

relation between the final values of the representatives (r_1 and r_2) and the parameters used to generate the dataset?

4. Instead of changing the value of the representatives for each example, accumulate the values of the difference $(x - r)$ and change the value only when all examples have been observed. Accumulate only for the closest representative in each iteration.

   for all x **do**

   $d \leftarrow d + (x - r)$

   end for

   $r \leftarrow r + (\alpha / n\_examples) * d$

5. Plot the consecutive positions of r_1 and r_2 and compare with the plot in exercise 1. What do you observe?

6. Plot with different colors:

   - color 1 – points closer to r_1 labeled 1;
   - color 2 – points closer to r_1 labeled 2;
   - color 3 – points closer to r_2 labeled 1;
   - color 4 – points closer to r_2 labeled 2.

   What do you observe?

7. Repeat the experiment 30 times and plot the final values of r_1 and r_2 over the dataset. If necessary amplify the viewed area to see the points' distributions.

Exercise 2

Implement a simplified version of agglomerative hierarchical clustering, as proposed in the following algorithm.

while there are more than two points **do**
    FIND the closest two points
    REPLACE both points by their average
end while

Test it on sets of points similar to the ones of the last exercise.

Exercise 3

Implement the DBScan algorithm as described in `https://www.youtube.com/watch?v=_A9Tq6mGtLI` and demonstrate graphically the process with a series of snapshots of the process at key points with adequate descriptions.