Laboratory Tutorial 12: Unsupervised Learning and K-Means Clustering algorithm

In this laboratory tutorial you will work with Unsupervised Machine Learning, specifically, with K-Means Clustering Algorithm.

Preamble

In this lab worksheet, you will perform k-means manually (to understand how it works) to group data points into clusters. You will then use scikit-learn library to run K-Means clustering algorithm in Python. You will need Jupyter notebook for Python 3.

Exercise 12.1: K-Means clustering algorithm manually

Consider the coordinates of the following points.

P1(2, 10)
P2(2, 5)
P3(8, 4)
P4(5, 8)
P5(7, 5)
P6(6, 4)
P7(1, 2)
P8(4, 9)

Using pen and paper you are required to perform k-means clustering, with k=3, determining which points belong to which of the three clusters. To do so:

i. Use the three points: (2,10), (5,8), (1,2) as initial cluster centers.
ii. Show all iterations (remember to stop when the means are not changing anymore) of the K-Means clustering algorithm. Additionally, at each iteration you must indicate which points belong to each cluster and the coordinates of the three new cluster centers.
iii. For each iteration you need to show graphically the points, and the mean that belong to each cluster.
iv. The distance between each pair of points must be calculated using Euclidean metric.

To calculate the distance, you can use one the following links:

https://www.calculatorsoup.com/calculators/geometry-plane/distance-two-points.php

https://www.translatorscafe.com/unit-converter/en-US/calculator/two-points-distance/

Exercise 12.2: K-Means clustering algorithm in Python (1)

We are going to run K-Means Algorithm to cluster our data using Python. Follow the steps:

1. Open Jupyter
2. Import the following libraries:

   ```
   import numpy as np

   from sklearn.datasets import make_blobs

   from sklearn.cluster import KMeans
   ```

3. Next, we need data. We will work with some random coordinates (let us go with 50), with k=2. The initial cluster means will be [(10,15), (4,7)]:

   ```
   num_of_coordinates = 50

   initial_cluster_means = [(10,15),(4,7)]

   number_of_clusters = len(initial_cluster_means)
   ```

4. We generate the data:

   ```
   dataset=make_blobs(n_samples=num_of_coordinates,centers=initial_cluster_means,n_features=2,cluster_std=2)
   ```

5. Display the data (50 random coordinates):

   ```
   random_data=dataset[0]

   print(random_data)
   ```

6. Create kmeans object:

   ```
   k=KMeans(n_clusters=number_of_clusters)

   k
   ```

7. We fit the k means object to the dataset and predict. Specifically, K-Means Algorithm is run on the 50 random coordinates, and group them in their respective cluster. We can now find out the coordinates that belong to each cluster.

   ```
   cluster_data = k.fit_predict(random_data)

   cluster_data
   ```

8. With the knowledge that you have gained in DAT1 do you think you can show graphically the points and the mean that belong to each cluster? Give it a go. If you are struggling ask me or the GTAs for help.

Exercise 12.3: K-Means clustering algorithm in Python (2)

You are required to run basic K-Means Algorithm in python to group the data points (those given in Exercise 12.1) into three clusters. Also, you will show graphically, the points, including the mean that belong to each cluster. You will need the code given in Exercise 12.3 to help you complete this exercise. However, there will be some minor changes in your code. For example, in Exercise 12.1 we gave K-Means the initial cluster means, but here we will let K-means to randomly pick three data points as the initial cluster means. Note that when you run basic K-Means only the last iteration, including the final clusters with their points and mean will be executed. Check if the last iteration that you had in exercise 12.1 is the same when you run K-Means in Python. Give it a go. If you are struggling ask me or the GTAs for help.

You should have the following output. **Note:** your output will only show the final iteration of the K-Means clustering algorithm.

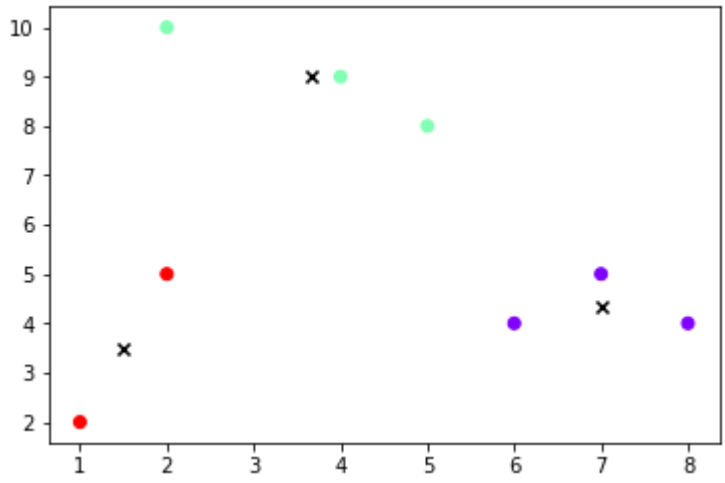
#Clusters and data points:

[1 2 0 1 0 0 2 1] # we have three clusters, denoted by 0, 1, and 2. Data point(2,10) belongs to cluster 1; data point (2,5) belongs to cluster 2, and so on.

#Cluster means:

```
[[7.         4.33333333]
 [3.66666667 9.        ]
 [1.5        3.5       ]]
```

#Graphically. Each cluster is denoted by a different colour. The means are represented by an x.



Summary

In this tutorial with regards to unsupervised learning, you have learned how to use K-Means Algorithm to cluster data. You have also gained knowledge of performing K-Means clustering algorithm using scikit-learn library in Python.

Reading

Textbook: Elements of Statistical Learning

- https://web.stanford.edu/~hastie/Papers/ESLII.pdf
- Chapter: 13
- Chapter: 14, Sections (14.3.6-14.3.7)