Autor: Susana Álvarez | susana.alvarezc@udea.edu.co

Lab II – Clustering

Machine Learning II

1. Research about the Spectral Clustering method, and answer the following questions:

a. In which cases might it be more useful to apply?

Spectral clustering excels in scenarios where the data showcases non-linear structures or more intricate patterns, such as concentric rings like the behavior of extreme temperatures in different seasons, or the distribution of different species in a habitat, including the relationships formed in a social network. In these instances, the spectral clustering algorithm proves to be particularly advantageous due to its capability to adeptly capture these nuanced situations, attributed to its ability to handle non-linear and complex relationships between data points.

b. What are the mathematical fundamentals of it?

The mathematical fundamentals of spectral clustering involve using the eigenvalues and eigenvectors of a similarity matrix derived from the data. The process typically includes constructing an affinity matrix, forming a Laplacian matrix, and then extracting the eigenvectors associated with the smallest eigenvalues. After extracting the eigenvectors, these vectors can be organized into a matrix, and the rows of this matrix can be treated as new feature representations of the original data points. Each row corresponds to a data point's projection onto the subspace defined by the selected eigenvectors.
Subsequently, a standard clustering algorithm, such as k-means, is often applied to these transformed data points. This helps in assigning the data points to distinct clusters based on their new representations in the eigenvector space.

c. What is the algorithm to compute it? The algorithm to compute spectral clustering generally involves the following steps:

1. Construct an affinity matrix to measure the similarity between data points.
2. Form a Laplacian matrix from the affinity matrix.
3. Compute the eigenvectors and eigenvalues of the Laplacian matrix.
4. Use the eigenvectors to form a new representation of the data.
5. Apply a standard clustering algorithm (e.g., k-means) to the transformed data to obtain the final clusters.

d. Does it hold any relation to some of the concepts previously mentioned in class? Which, and how?

Spectral clustering is closely related to concepts like eigenvectors, k-means, and k-median. Let's explore these relationships:

- Eigenvectors: Spectral clustering relies on the eigenvalues and eigenvectors of the Laplacian matrix derived from the data. The eigenvectors associated with the smallest eigenvalues are crucial for defining a new representation of the data in a lower-dimensional space. These eigenvectors capture the intrinsic structure of the data, enabling more effective clustering.

- K-means, kmedoids: Spectral clustering frequently incorporates k-means as a subsequent step. Following the extraction of eigenvectors, a standard clustering algorithm such as k-means is employed on these vectors. The data points, now represented by the eigenvectors, are then clustered using k-means to delineate distinct groups. This amalgamation empowers spectral clustering to effectively manage non-linear and intricate structures within the data. Furthermore, it can be coupled with k-medoids, an alternative clustering algorithm that proves beneficial when handling noisy or outlier-prone data.

2. Research about the DBSCAN method, and answer the following questions:

   a. In which cases might it be more useful to apply? In which cases might it be more useful to apply?
   Like spectrum clutering DBSCAN is useful when the data has irregularity shaped and have variaty densities. DBSCAN handling noise data and oitlaiers effectly.

   b. What are the mathematical fundamentals of it? The core idea of DBSCAN is based on the concept of density reachability. Here are the key components:

   Density Reachability: A point A is density-reachable from point B if there is a chain of points p1,p2,...,pnp1,p2,...,pn such that p1=Bp1=B, pn=Apn=A, and each pi+1pi+1 is directly density-reachable from pipi.

   Density Connectivity: Two points A and B are density-connected if there is a point C such that both A and B are density-reachable from C.

   Core Points, Border Points, and Noise: DBSCAN categorizes points into core points (high-density points), border points (reachable from core points but not dense enough to be core), and noise points (neither core nor border).

   c. Is there any relation between DBSCAN and Spectral Clustering?
   Indeed, both DBSCAN and Spectral Clustering are clustering algorithms. DBSCAN exhibits versatility and robustness in handling scenarios involving noise and variable densities. On the other hand, Spectral Clustering is better suited for datasets featuring

intricate, non-linear structures, particularly when these structures can be effectively represented as a graph. In essence, while both algorithms serve the purpose of clustering, they excel in different contexts.

3. What is the elbow method in clustering?

   The elbow method is a graphical technique employed in clustering, specifically for determining the optimal number of clusters (K) in algorithms like k-means. It entails plotting a metric, such as the within-cluster-sum-of-square (WCSS) or distortion, against varying values of K. The graph exhibits a trend where the metric initially decreases as K increases, reaching a point where the rate of decrease notably slows down, forming an elbow-like bend. This elbow point signifies the optimal K value, reflecting a balance between capturing data variance and avoiding unnecessary complexity in cluster formation. In summary, the elbow method provides a visual aid in identifying the most suitable number of clusters for a given dataset.

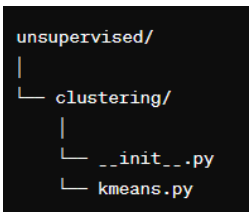
   And which flaws does it pose to assess quality?

   - Dependence on Data Distribution: The effectiveness of the elbow method is influenced by how data points are distributed within the dataset. In instances where clusters exhibit irregular shapes or varying densities, the identification of a distinct elbow point in the graph may be challenging.

   - Does Not Always Indicate Optimal Clusters: The elbow method may not always lead to the identification of the optimal number of clusters. In some cases, the elbow may not be pronounced, or there might be multiple bends in the curve.

4.

   a. Implement the k-means module using Python and Numpy. The package path can be found at the following location:



   b. Implement the k-medoids module using Python and Numpy: The package path of algorithm can be found at the following location:

```
unsupervised/
|
└── clustering/
    |
    └── __init__.py
    |
    └── KMedoids.py
```

c. Both of algorhitms use:

- Constructor Initialization that follows the convention of setting parameters with default values, allowing users to instantiate the `KMedoids` or Kmeans class with specified or default parameters,

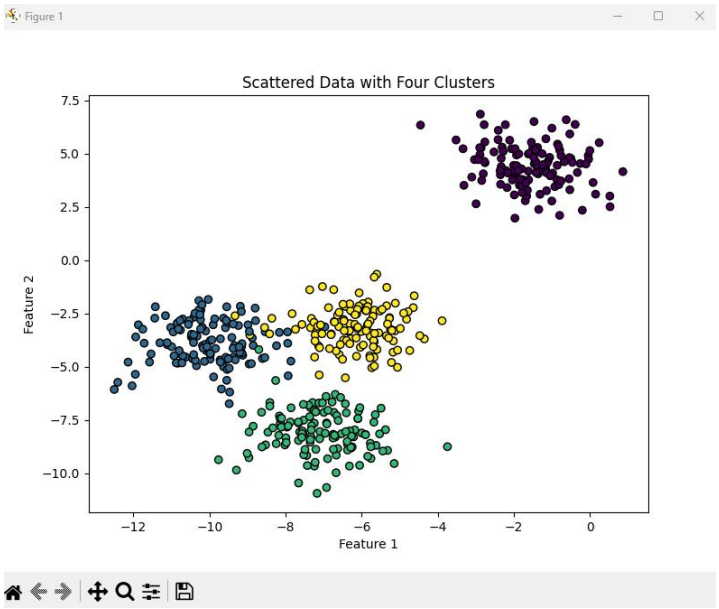
- **fit** method is consistent with Scikit-Learn's convention. It takes a dataset `X` as input, performs the clustering, and updates the internal attributes (`labels_` and `cluster_centers_`).

5.

b. Plot the resulting dataset.

c.

How many clusters are there? In the graph, it is evident that there are 4 clusters, The choice of dividing the data into 4 clusters is evident not only through visual inspection but also by the explicit setting of the `centers` parameter to 4 in the `make_blobs` function.



How far are they from one another? The purple cluster stands out as notably distant from the other groups. In contrast, the yellow, green, and blue clusters are closely

situated, suggesting a higher degree of proximity and potential similarity in their respective data points.

The package path of this excercise can be found at the following location

\unsupervised\5b.py

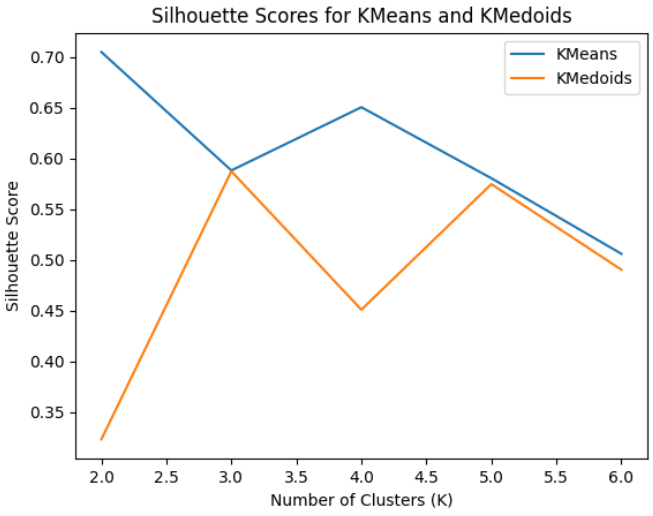
d. For both k-means and k-medoids (your implementations), calculate the silhouette plots and coefficients for each run, iterating K from 1 to 5 clusters.

The package path of this excercise can be found at the following location

\unsupervised\silhouette.py

e. What number of K got the best silhouette score? The highest silhouette score for KMeans is 2 and KMedoids occurs when the number of clusters (K) is set to 3.

```
kmeans   Cluster 2 puntaje:0.7049787496083262
kmedoids Cluster 2 score :0.3229932624109135
kmeans   Cluster 3 puntaje:0.5882004012129721
kmedoids Cluster 3 score :0.5873430979447513
kmeans   Cluster 4 puntaje:0.6505186632729437
kmedoids Cluster 4 score :0.4508628263748775
kmeans   Cluster 5 puntaje:0.580497634034948
kmedoids Cluster 5 score :0.5746932321727457
kmeans   Cluster 6 puntaje:0.5059890580233962
kmedoids Cluster 6 score :0.49029520333216825
```



What can you say about the figures? The silhouette score is relatively high when K is set to 2 for Kmeans and 3 for kmedoids, indicating well-defined clusters, however, as K increases beyond 2, the silhouette score decreases, suggesting a decrease in cluster

quality. The trend aligns with the expected behavior, as a higher number of clusters might lead to less cohesive and more overlapping clusters.

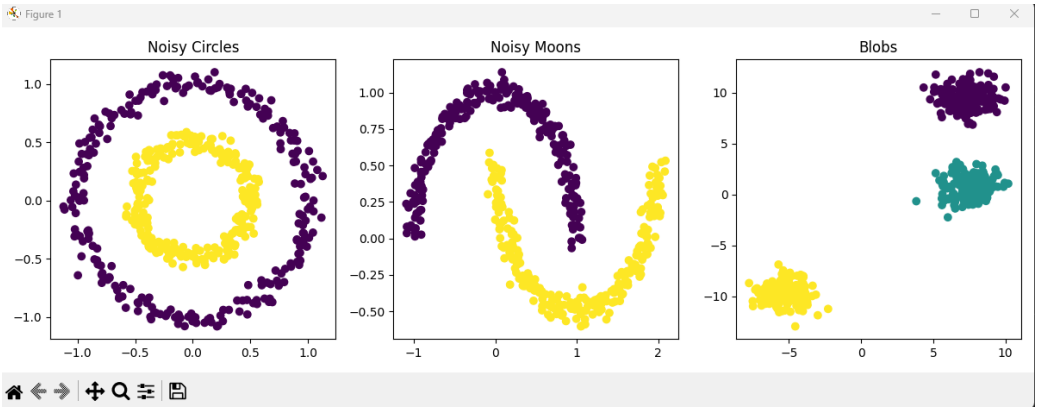
6.

a. Plot the different datasets in separate figures.

The package path of this excercise can be found at the following location

\unsupervised\6clustering.py

b. What can you say about them?



it's composed of tre different structures: two circles, two moons, and a three blobs, each with a different density.

In the graph "Noisy Circles" the function make_circles generates points arranged in concentric circles. The scatter plot illustrates these circles, and different colors signify distinct clusters. The presence of noise introduces some variability in the points.

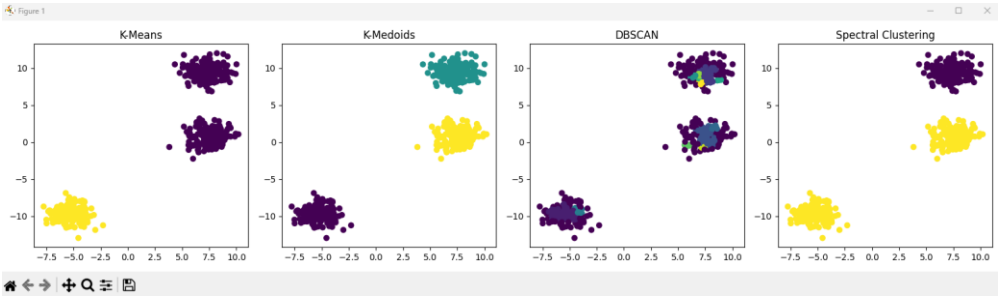
In graph "Noisy moons" we observe a figure in the plane forming two opposite moons.

In the graph "Blobs", we are generating 500 points distributed across three blobs and plotting them. The various colors in the scatter plot indicate different clusters. The blobs exhibit uniform dispersion and maintain a standard deviation of 1.0.

Apply k-means, k-medoids, DBSCAN and Spectral Clustering from Scikit-Learn over each dataset and compare the results of each algorithm with respect to each dataset.

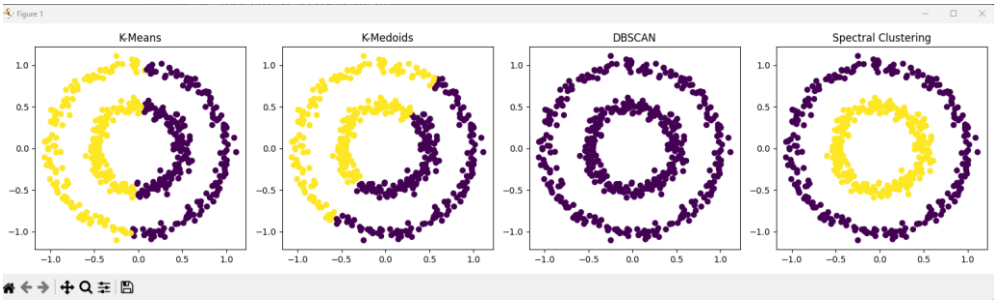
- Blobs dataset

- Spectral Clustering seems to have identified two out of the three clusters.
- K-Means shows an imbalance in the distribution of points between the two clusters.
- K-Medoids has successfully identified all three clusters in a more balanced manner.
- DBSCAN has detected multiple clusters, but some of them might be considered as noise or very small groupings.



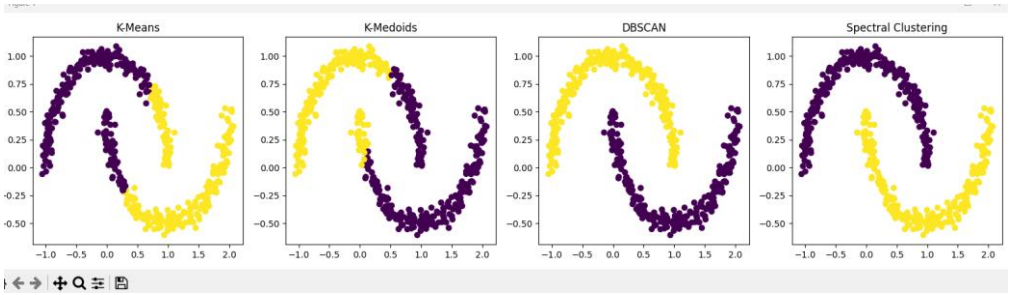
- Noisy Circles Dataset:

As expected, the best-performing model that clustered the data was Spectral Clustering since it is less constrained by assumptions about cluster shapes compared to methods like K-Means. It can adapt to the inherent structure of the data, making it more flexible in capturing circular or non-linear patterns.



- Noisy Moons Dataset:

The Noisy Moons dataset has a non-convex distribution with two crescent shapes. DBSCAN and Spectral Clustering are known for their ability to handle non-convex structures, allowing them to effectively capture the crescent moons



K-Means and K-Medoids are algorithms designed to find convex clusters. When clusters have non-convex shapes, such as in the case of crescent moons, these algorithms may struggle to properly separate the groups. K-Means can split a moon into two, and K-Medoids may be affected by the presence of outliers.