**Workshop II**

**Camilo Llerena Riascos**

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

1. In which cases might it be more useful to apply?
2. What are the mathematical fundamentals of it?
3. What is the algorithm to compute it?
4. Does it hold any relation to some of the concepts previously mentioned in class? Which, and how?

**a.** Spectral clustering is a powerful technique used in unsupervised learning and data clustering. It has several advantages over other clustering algorithms like K-means, hierarchical clustering, and DBSCAN.

Spectral clustering is particularly useful when:

* The data has non-linear decision boundaries: Spectral clustering can be used to identify complex non-linear patterns in data that cannot be easily identified by other clustering algorithms.
* The data has high dimensions: Spectral clustering can handle high-dimensional data effectively by reducing the dimensionality of the data using techniques like principal component analysis (PCA) or t-SNE.
* The data has clusters with different shapes and sizes: Spectral clustering can identify clusters of arbitrary shapes and sizes, making it useful in a wide range of applications.
* The data has noisy or missing values: Spectral clustering can handle noisy and missing data by using graph Laplacian regularization and can still produce meaningful clusters.
* The data has an underlying graph structure: Spectral clustering works well when the data can be represented as a graph or network.

In summary, spectral clustering is a powerful technique that can be applied in various scenarios where the data is high-dimensional, non-linear, and has complex structures.

**b.** Spectral clustering is based on the spectral graph theory, which relates the eigenvalues and eigenvectors of a graph Laplacian matrix to the geometric properties of the data. The Laplacian matrix is a square matrix that is constructed from the data matrix, where each row of the Laplacian corresponds to a data point in the dataset.

There are two types of Laplacians that are commonly used in spectral clustering:

1. Unnormalized Laplacian: It is defined as L = D - A, where D is the diagonal matrix of the degree of each node in the graph, and A is the adjacency matrix of the graph. The unnormalized Laplacian is a real and symmetric matrix, which can be diagonalized using eigendecomposition.
2. Normalized Laplacian: It is defined as L = I - D^-1/2 \* A \* D^-1/2, where I is the identity matrix, D^-1/2 is the diagonal matrix of the inverse square root of the degree of each node, and A is the adjacency matrix of the graph. The normalized Laplacian is also a real and symmetric matrix, which can be diagonalized using eigendecomposition.

The eigenvectors of the Laplacian matrix are used to perform clustering in spectral clustering. The k smallest eigenvalues and their corresponding eigenvectors are selected, where k is the number of clusters to be identified. The eigenvectors are then used to construct a new low-dimensional feature space, where the data points are projected. Finally, a clustering algorithm like K-means is applied to cluster the projected data points.

The intuition behind spectral clustering is that the low-dimensional embedding space captures the underlying structure of the data and separates the different clusters effectively. In particular, the eigenvectors of the Laplacian matrix correspond to the smoothest functions on the graph, and the kth eigenvector can be seen as encoding information about the kth smallest cluster. Spectral clustering leverages this property to identify clusters in the data.

**c.** The Spectral Clustering algorithm using the Scikit-learn library:

import numpy as np

from sklearn.cluster import SpectralClustering

from sklearn.metrics import pairwise\_distances

# Step 1: Construct a similarity matrix W based on the dataset X

X = ... # input data

W = pairwise\_distances(X, metric='euclidean')

# Step 2: Construct a graph from the similarity matrix W

graph = ... # construct a graph from the similarity matrix W

# Step 3: Compute the Laplacian matrix L, either the unnormalized or normalized Laplacian

L = ... # compute the Laplacian matrix L

# Step 4: Compute the k smallest eigenvectors and their corresponding eigenvalues of the Laplacian matrix L

n\_clusters = ... # number of clusters to identify

eigenvalues, eigenvectors = np.linalg.eigh(L)

indices = np.argsort(eigenvalues)[:n\_clusters]

Y = eigenvectors[:, indices]

# Step 5: Form the matrix Y by concatenating the k eigenvectors

Y = np.hstack([Y[:, i] for i in range(n\_clusters)])

# Step 6: Normalize the rows of the matrix Y

norms = np.linalg.norm(Y, axis=1)

Y\_normalized = Y / norms[:, np.newaxis]

# Step 7: Apply the clustering algorithm to the matrix Y to obtain k clusters

clustering = SpectralClustering(n\_clusters=n\_clusters, affinity='nearest\_neighbors')

clustering.fit(Y\_normalized)

# Step 8: Assign labels to each data point based on the cluster it belongs to

labels = clustering.labels\_

# Step 9: Output the clustering labels

print(labels)

**d.** Relation to Spectral Clustering method

1. Spectral clustering and principal component analysis (PCA) are two different methods that are often used in machine learning and data analysis. However, there is a relationship between them. PCA is a technique for reducing the dimensionality of data by identifying the most important features or components that capture the most variation in the data. It does this by computing the eigenvectors and eigenvalues of the covariance matrix of the data and using them to transform the data into a lower-dimensional space. Spectral clustering, on the other hand, is a method for clustering data by identifying groups of points that are similar to each other based on their pairwise similarity or distance. It does this by computing the eigenvectors and eigenvalues of a similarity or Laplacian matrix and using them to transform the data into a lower-dimensional space, where clustering can be performed more easily.

The relationship between PCA and spectral clustering is that they both involve computing the eigenvectors and eigenvalues of a matrix. In fact, the first step of spectral clustering often involves performing PCA on the data to reduce its dimensionality before computing the Laplacian matrix. This is because the Laplacian matrix is typically high-dimensional and sparse, and performing PCA can help to reduce its dimensionality and sparsity.

1. Spectral clustering and SVD are two different methods that can be used together in machine learning and data analysis. SVD can be used as a preprocessing step for spectral clustering, or to perform dimensionality reduction and feature extraction before clustering with spectral clustering. Both methods involve decomposing a matrix into its constituent parts, and they can be used to extract useful information from data for clustering and analysis.
2. t-SNE and spectral clustering are two different methods that can be used together in machine learning and data analysis. t-SNE can be used as a preprocessing step for spectral clustering, or to visualize the data before clustering with spectral clustering. Both methods involve transforming high-dimensional data into a lower-dimensional space, where clustering or visualization can be performed more easily.

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

1. In which cases might it be more useful to apply?
2. What are the mathematical fundamentals of it?
3. Is there any relation between DBSCAN and Spectral Clustering? If so, what is it?

**a.** DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm in machine learning and data mining. It is a density-based clustering algorithm that groups together data points that are closely packed together, while also identifying points that are outliers or noise.

DBSCAN can be useful in several cases, including:

* When the data has a high degree of noise and outliers: DBSCAN can effectively handle noisy and sparse datasets, as it is able to identify and ignore points that are not part of any cluster.
* When the data has complex or irregular shapes: DBSCAN is able to identify clusters of any arbitrary shape and is not limited to spherical clusters, which can be a limitation of other clustering algorithms like k-means.
* When the number of clusters is not known beforehand: DBSCAN does not require the number of clusters to be specified in advance, which can be a significant advantage over other clustering algorithms.
* When the data is not uniformly distributed: DBSCAN can handle datasets where the density of points varies throughout the dataset, which can be a limitation of other clustering algorithms that assume uniform distribution.
* When the data has a large number of dimensions: DBSCAN is relatively insensitive to the curse of dimensionality, which can be a limitation of other clustering algorithms that become less effective as the number of dimensions increases.

**b.** DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups together data points that are closely packed together while also identifying points that are outliers or noise. The mathematical fundamentals of DBSCAN include:

1. Density: The core concept of DBSCAN is the idea of density. Density can be defined as the number of points within a specified distance of a given point. In DBSCAN, a point is considered a core point if there are at least a minimum number of points within a specified distance (epsilon) of it.
2. Epsilon: Epsilon (ε) is a distance threshold that is used to determine the neighborhood of a point. A point is considered to be within the neighborhood of another point if it is within a distance of ε from that point.
3. Core Points: A point is considered a core point if there are at least a minimum number (MinPts) of points within its ε-neighborhood.
4. Border Points: A point is considered a border point if it is not a core point but lies within the ε-neighborhood of a core point.
5. Noise Points: A point is considered a noise point if it is neither a core point nor a border point.
6. Reachability: DBSCAN defines a notion of reachability between points. A point p is said to be reachable from another point q if there exists a path of core points from q to p such that each successive core point is within the ε-neighborhood of the previous core point.
7. Clusters: A cluster in DBSCAN is defined as a set of core points that are reachable from each other. A cluster can also include border points that are within the ε-neighborhood of a core point.
8. The algorithm works by first identifying core points, then expanding clusters by including border points that are within the ε-neighborhood of a core point. Points that are not part of any cluster are considered noise points. The algorithm is sensitive to the parameters ε and MinPts, which can be adjusted to control the clustering results.

**c.** Despite their differences, there is a relation between DBSCAN and Spectral Clustering in that they can both be used for clustering data with complex structures. Spectral Clustering can be effective when the data has a clear underlying graph structure, while DBSCAN can be more effective when the data has varying densities or is not uniformly distributed. In some cases, Spectral Clustering can be used as a pre-processing step to generate a similarity matrix that is then used as input to DBSCAN to cluster the data.

**3. What is the elbow method in clustering? And which flaws does it pose to assess quality?**

The elbow method is a common approach to determine the optimal number of clusters in a dataset for clustering algorithms. The method involves plotting the within-cluster sum of squares (WCSS) against the number of clusters. The WCSS measures the sum of the squared distances between each data point and the centroid of its assigned cluster. The idea is to choose the number of clusters at which the rate of decrease in WCSS slows down, forming an "elbow" in the plot. The number of clusters at the elbow point is considered the optimal number of clusters.

While the elbow method is simple and easy to implement, it has some flaws that can lead to incorrect or suboptimal clustering results. One of the main issues is that the elbow point is often subjective and difficult to determine, especially for datasets with complex structures or overlapping clusters. Additionally, the elbow method may not always capture the true underlying structure of the data and can lead to over- or under-segmentation. Another issue is that the method assumes that the WCSS is a good measure of clustering quality, which may not always be the case, particularly for non-convex clusters or high-dimensional data. Therefore, it is important to use the elbow method in conjunction with other techniques and to carefully evaluate the clustering results to ensure their quality.

**5.b. Plot the resulting dataset. How many clusters are there? How far are they from one another?**

There are 4 well defined clusters and the graph shows that one cluster is distant from the other.

**5. d. What number of K got the best silhouette score? What can you say about the figures? Is this the**

**expected result?**

The best silhouette score in kmeans is 0.7049787496083262 for clusters.

The best silhouette score in kmedoids is 0.7049787496083262

This result is expected. Not necessarily. Increasing the number of clusters in a clustering algorithm may improve the accuracy of data clustering in some cases, but in others it may have no effect or even worsen the quality of the clustering.

As the number of clusters is increased, each cluster will have fewer data points, which can make it more difficult to identify meaningful patterns in the data.

**6. a. Plot the different datasets in separate figures. What can you say about them?**

There are 6 types of distribution with different characteristics, for example non-linear such as Noisy Circles and Noisy Moon, contain overlapping sets, which makes it difficult to separate by clustering techniques. Blobs have a uniform distribution in the region of space in which they are found, and Anisotropy is a dataset that has clusters that follow a band structure.

On the other hand, unstructured distributions such as Varied Variances and No Structure a priori do not have a definition of clustering.

**References**

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In Advances in neural information processing systems (pp. 849-856). <https://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf>

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96) (pp. 226-231). <https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf>