Spectral Clustering Problem

Ahmad Sidani
Politecnico di Torino
S312919
ahmad.sidani@studenti.polto.it

Ali Yassine
Politecnico di Torino
S312920
ali.yassine@studenti.polto.it

Hadi Ibrahim
Politecnico di Torino
S313385
hadi.ibrahim@studenti.polto.it

***Abstract*—In machine learning and data analysis, spectral clustering [1] is a prominent method for clustering datasets based on their underlying structure. In this study, we examine how spectral clustering is applied to two different datasets, Circle and Spiral, and compare the findings to other clustering algorithms**

I. Introduction

Spectral clustering is an effective unsupervised learning approach that has grown in popularity in recent years. It is based on spectral graph theory and enables an approach to cluster data points in a high-dimensional space by projecting them onto a lower-dimensional space with the eigenvectors of a similarity graph's Laplacian matrix. This method has been demonstrated to be useful in a variety of applications, including image segmentation, community detection, and document clustering.

In this report, we will evaluate the performance of spectral clustering in clustering data points on two data sets, Circle.csv and Spiral.csv. We will implement spectral clustering in a step-by-step manner, beginning with the generation of the similarity graph using a Gaussian kernel. The Laplacian matrix will then be computed, and its eigenvectors will be used to conduct clustering.

We will also compare the results with other clustering methods, such as Agglomerative clustering [2], a form of hierarchical clustering algorithm that divides each data point into its own cluster before merging the clusters until a stopping condition is satisfied. Finally we will discuss the benefits and drawbacks of spectral clustering.

II. Datasets

In this research, we will use two datasets: Circle.csv and Spiral.csv. The datasets include N points and are in CSV format.

The Circle.csv file has two columns that correspond to the x and y values of the points. The dataset is made up of a random distribution of points that lies on two circles, with some noise.

The Spiral.csv dataset has three columns: the first two correspond to the x-values and y-values of the points, while the third column includes the index of the proper cluster. The dataset is made up of points on three intertwined spirals.

Both datasets will be utilized to generate a similarity graph, calculate the Laplacian matrix, and carry out spectral clustering. To evaluate the performance of spectral clustering on these datasets, we will compare the findings with other clustering approaches such as k-means.

For better understanding, Figure 1 below visualizes both datasets.

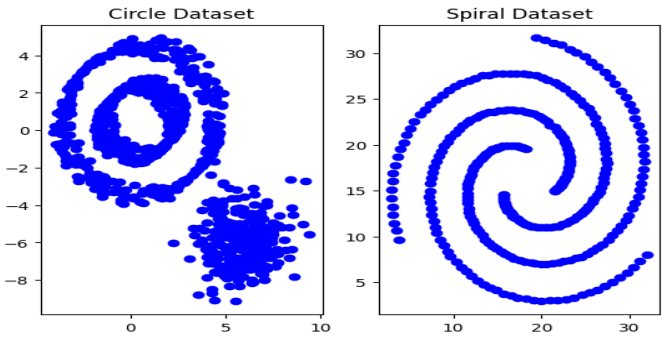

Fig. 1. Comparison of Datasets Using Scatter Plots.

III. Methodology

In this section, we will go through the methodology utilized to conduct the spectral clustering. The methodologies below were applied to both the Spiral and Circle datasets.

A. Construction of Similarity Graph

A similarity function that determines the degree of similarity between each pair of points is used to build the similarity graph. The exponential function with a Gaussian kernel, which is specified as the similarity function in this project, is defined as:

$$s_{i,j} = \exp\left(-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{2\sigma^2}\right)$$

B. Degree & Laplacian Matrices

1) Degree Matrix: The degree of each graph vertex is represented by a diagonal matrix. The degree of vertex i is specifically represented by the diagonal entry of the matrix at position (i,i) where the degree of a vertex is the number of edges incident to that vertex. The degree matrix D is a symmetric matrix for an undirected graph.

The degree matrix D is calculated using the following formula:

$$D_{i,i} = \text{degree}(i)$$

where i is the vertex's index and degree(i) is the vertex's degree. The weights of the edges that are incident on a vertex are simply added to determine its degree. The degree matrix D is a n × n diagonal matrix for an undirected graph with n vertices, where the i-th diagonal element of D is determined by the degree of vertex i. In notation for mathematics, we can write:

$$D = \text{diag}(\text{degree}(1), \text{degree}(2), \ldots, \text{degree}(n))$$

2) Laplacian Matrix: A matrix that encodes crucial structural details about a graph is known as the Laplacian matrix. According to the following definition, it is the graph's difference between the degree matrix and adjacency matrix:

$$L = D - W$$

where L is the Laplacian matrix, D is the degree matrix, and W is the adjacency matrix.

The Laplacian matrix L is a symmetric positive semi-definite matrix, which means that its eigenvalues are not negative. The degree of each vertex is represented by the diagonal elements of the Laplacian matrix, and the number of edges connecting adjacent vertex pairs is represented by the off-diagonal entries.

3) Conversion to sparse format: The Laplacian matrix L, Degree matrix D, and Adjacency matrix W were changed into sparse format to reduce memory usage and increase computing efficiency. Just the non-zero entries of these matrices are stored in memory when they are represented in a sparse format, which lowers the amount of memory needed to store the matrices. This is crucial for huge matrices because they can have excessively high memory requirements for a dense format. Furthermore, sparse matrices make computations more effective by minimizing unnecessary operations on zero entries.

C. Connected Components

The Laplacian matrix is a crucial tool for investigating graph connectivity, and its spectrum offers insightful data about the graph's composition. We can gain a sense of the dataset's structure and identify any unconnected subgroups by counting the number of linked components. We discovered that there were two connected components in the Circle.csv dataset, compared to just one in the Spiral.csv dataset.

In order to optimize the clustering algorithm's performance, connected component analysis is crucial. Smaller clusters can be formed when there are a lot of connected components, but larger clusters can be formed when there are few connected components.

D. Choosing Number of Clusters

A critical step in spectral clustering [1] is calculating the number of clusters. It is based on the Laplacian matrix's eigenvalues and eigenvectors. Eigenvalues are the scaling factors for eigenvectors, which are vectors that remain in the same direction when multiplied by a matrix but are scaled by a constant factor. The Laplacian matrix is the matrix for which the eigenvectors and eigenvalues are computed in the context of spectral clustering. The number of clusters in the data is indicated by the small eigenvalues of the Laplacian matrix. In our case we used Arnoldi technique to compute the K-smallest eigenvalues and their related eigenvectors. The Arnoldi technique [3] is a Krylov subspace method that excels in computing a few eigenvalues of a large, sparse matrix. To figure out how many clusters there are, we plot small number of eigenvalues in ascending order and look for the 'elbow' in the plot. The elbow denotes the point at which the eigenvalues sharply decrease, suggesting the optimal number of clusters. For our analysis, we implemented it in the circle and spiral datasets. The plot for the circle dataset is shown in Figure 2. The graphic demonstrates that the eigenvalues grow sharply at 3, indicating that the data can be divided into three clusters. Figure 3 is the eigenvalues plot for the spiral dataset, which similarly reveals a strong increase in eigenvalues at 3. This suggests that the data can also be divided into three clusters.

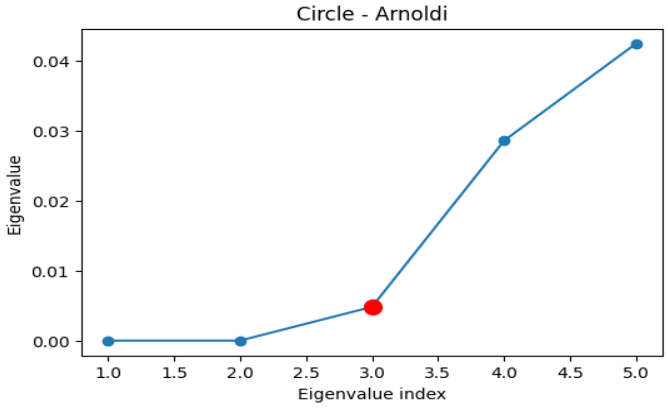

Fig. 2. The ascending order of eigenvalues for the Laplacian matrix of the Circle dataset.

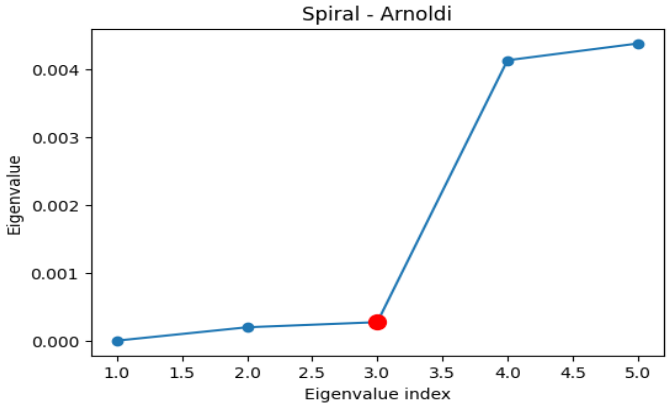

Fig. 3. The ascending order of eigenvalues for the Laplacian matrix of the Spiral dataset.

E. Running K-Means Algorithm

In this part, we build the sparse matrix U for each, whose columns are the eigenvectors u_1,..., u_M. We define y_i $\mathbb{R}^M$ as the vector corresponding to the i-th row of U for i = 1,..., N. The points in $\mathbb{R}^M$ with y_i, i = 1,..., N are then clustered using the k-means technique into clusters C_1,..., C_M. By minimizing the sum of the squared distances between each data point and the centroid of its cluster, the iterative k-means method divides a set of data points into k clusters. Each original point in the original Spiral and Circle datasets is given its corresponding cluster once the k-means algorithm has been executed on both of them.

IV. Results

In this section we demonstrate the outcomes of applying Spectral Clustering and Agglomerative Clustering to the Circle and Spiral datasets.

A. Results of Spectral Clustering

For both datasets, Spectral Clustering generated excellent results with distinct cluster separations. Figure 4 shows how Spectral Clustering correctly distinguished the two various circles for the Circle dataset. Figure 5 illustrates how Spectral Clustering successfully distinguished the spirals in the Spiral dataset from one another.

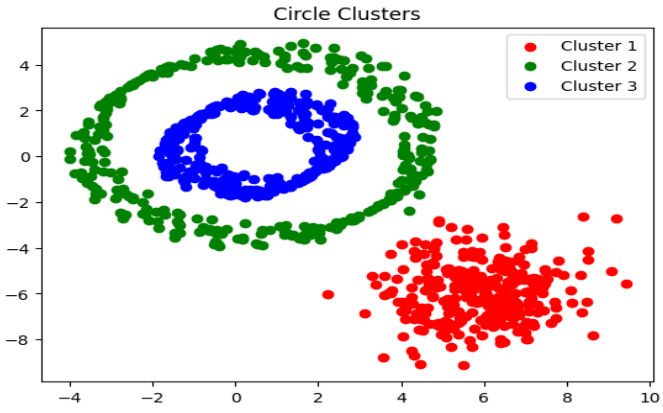

Fig. 4. Scatter-plot of the Circle dataset that has been clustered into three clusters using the spectral clustering algorithm.

B. Results of Agglomerative Clustering

Agglomerative Clustering was able to identify some differences in the data, but it was unable to differentiate the clusters where it mixed some clusters together. This is seen in Figures 6 and 7, which clearly reveal that there are different clusters than those found using Spectral Clustering.

C. Results of Comparison of Results

In this part, visual analysis of the spiral and circle datasets and an adjusted Rand index comparison for the spiral dataset were used to assess the effectiveness of spectral clustering and agglomerative clustering.

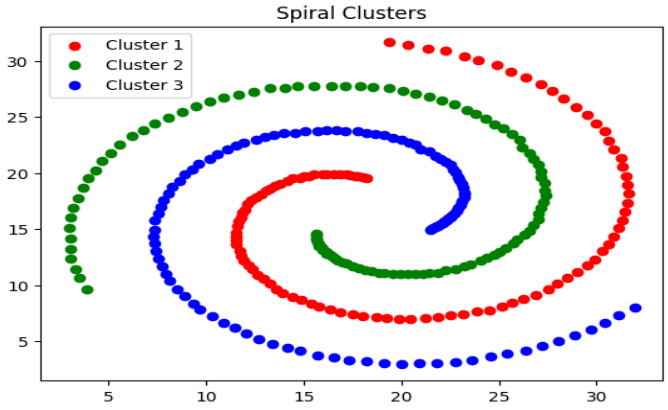

Fig. 5. Scatter-plot of the Spiral dataset that has been clustered into three clusters using the spectral clustering algorithm.

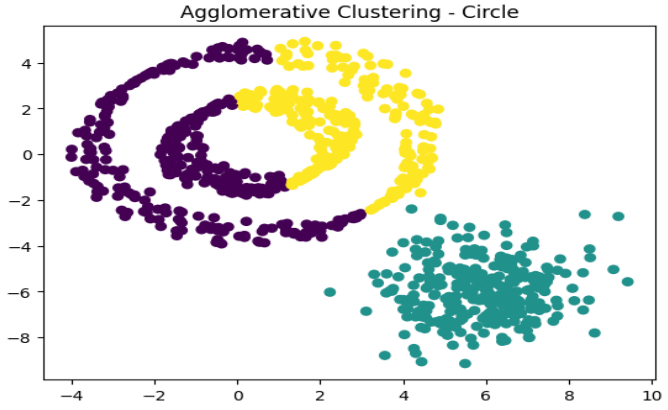

Fig. 6. Scatter-plot of the Circle dataset that has been clustered into three clusters using the agglomerative clustering algorithm.

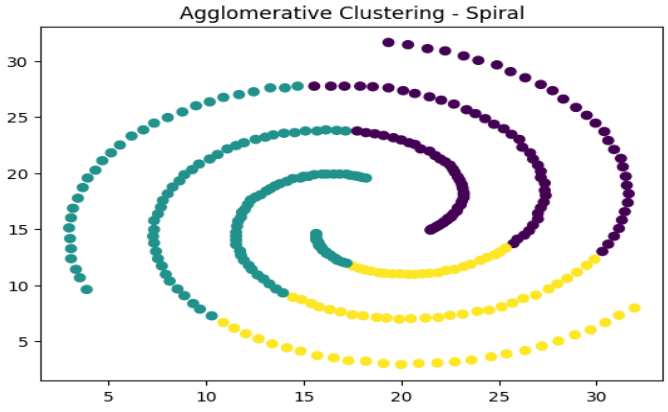

Fig. 7. Scatter-plot of the Spiral dataset that has been clustered into three clusters using the agglomerative clustering algorithm.

Visually we can clearly identify that the two circles in the circle dataset could be distinguished from one another and from the noise using spectral clustering, whereas the clusters were merged and differentiation the two circles wasn't obtained using agglomerative clustering. Similarly, spectral clustering, as opposed to agglomerative clustering, was able to recognize each spiral in the spiral dataset.

The adjusted Rand index that ranges between -1 and 1 that compares the predicted labels with the true labels gave a value of 1.0 for spectral clustering and -0.0008805 for agglomerative clustering indicating that spectral clustering provided more accurate clustering with a perfect score.

V. Discussion

In this study, we compared the performance of two unsupervised learning approaches , Spectral clustering and Agglomerative clustering on two datasets. We provided a detailed explanation of the methodology we adopted, which involved creating the degree matrix, computing the Laplacian matrix, and then converting it to a sparse format before using its eigen vectors to construct a matrix that was used in the k-means algorithm.

Evidently, spectral clustering outperformed agglomerative clustering based on the visual analysis and the adjusted rand index. This proves that the available data points were clustered more precisely using spectral clustering than by agglomerative clustering. One of the explanations for this is that complex data, like the Spiral dataset, are better suited for spectral clustering since it can capture the underlying geometric structure of the data.

This report emphasizes how well spectral clustering clusters handles complicated geometrical data. For data analysts who want to find hidden structures in their data, it is a helpful and essential tool. In addition to this, analysts may better comprehend their data and produce insightful conclusions by using spectral clustering to help with pattern recognition and group formation.

Future research should investigate other clustering techniques, as agglomerative clustering might not be appropriate for complicated geometrical data. The optics algorithm [4], a density-based clustering technique that can handle complicated geometries and outliers, is one possible substitute. Another choice is deep learning-based clustering techniques, which may perform better than conventional clustering techniques. These techniques include autoencoders [5] and self-organizing maps [6], which may automatically understand the underlying structure of the data.

References

[1] Wang, F., Zhang, C., Liu, C., Liu, Z., Zhang, X. (2020). "Spectral clustering: a review and perspectives." Frontiers of Computer Science, 14(5), 865-890.

[2] Tian, Y., Bai, X. (2021). "An Adaptive Agglomerative Clustering Framework for Multi-view Spectral Clustering." IEEE Transactions on Cybernetics, 51(2), 863-874.

[3] Baglama, J., Reichel, L. (2020). ARPACK software for large scale eigenvalue problems: Latest developments. Journal of Computational and Applied Mathematics, 375, 112594.

[4] Ankerst, Mihael, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. "OPTICS: ordering points to identify the clustering structure." ACM Sigmod Record 28, no. 2 (1999): 49-60. https://dl.acm.org/doi/10.1145/304181.304187

[5] Hinton, G. E., Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

[6] Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE, 78(9), 1464-1480.