Technical Report on Spectral Clustering

Arian Mohammadi s346278
Mustafa Kerem Kose s339018

Abstract

This document presents the application and assessment of spectral clustering techniques in three datasets which are the Circle, Spiral, and Sphere (a 3D dataset). The methods include the construction of similarity graphs, derivation of two types of Laplacian matrices-normalized and unnormalized- computation of eigenvalues and eigenvectors, and clustering them for the data points. The results of spectral clustering are compared with other methods such as K-means, hierarchical and DBscan clustering to analyze their efficiency over the datasets.

1 Introduction

Spectral clustering is a powerful technique for the main meaning of grouping data points according to the structure of a similarity graph. It uses the eigenvalues as well as the eigenvectors of graph Laplacians for intrinsic patterns in data. The present work is concerned with broadening the understanding of spectral clustering to three datasets: Circle, Spiral, and Sphere (a 3-dimensional dataset). The study of two different types of Laplacian matrices: admissible normalized and unnormalized and each with its different characteristics to interpret the structure of the data. This work discusses those compulsory topics in the homework with extra details on the efficiency of using spectral clustering on these datasets.

2 Implementation Details

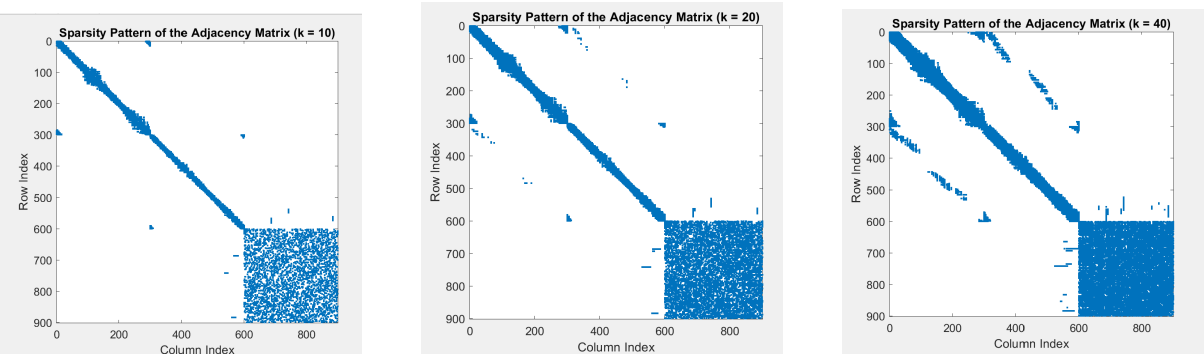
Dataset and Similarity Graph Construction

Data from `Circle.csv`, `Spiral.csv`, or `Sphere.csv` is loaded according to a selection made by a user. Each dataset related to a particular clustering challenge: Circle and Spiral have 2D shapes, while Sphere has a 3D shape.

A k-nearest neighborhood graph is created with the help of a Gaussian similarity function:

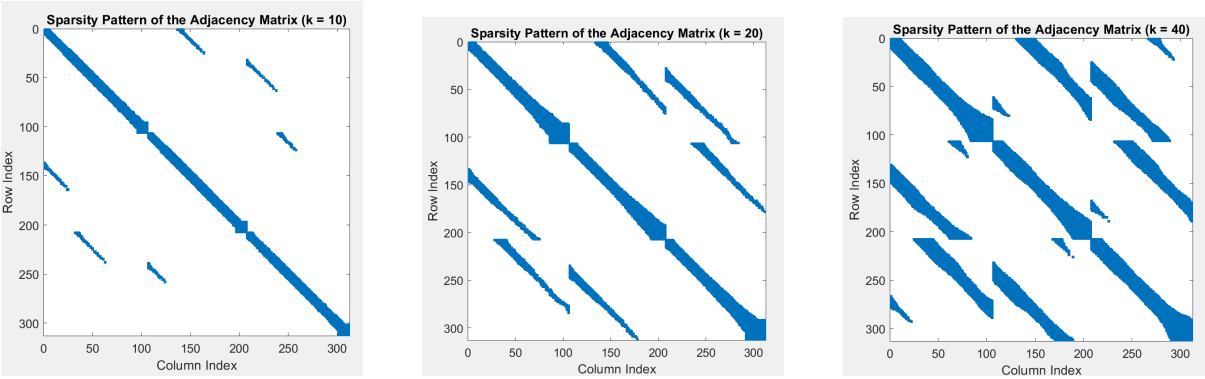
$$s_{i,j} = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right).$$

Here, σ is fixed to 1, while different varieties of k (10, 20, 40) have been taken to study the effect of neighborhood sizes on clustering. Then an adjacency matrix has been constructed and visualized in order to analyze the relationships between the data points and the built structure of the similarity graph for the data returned.



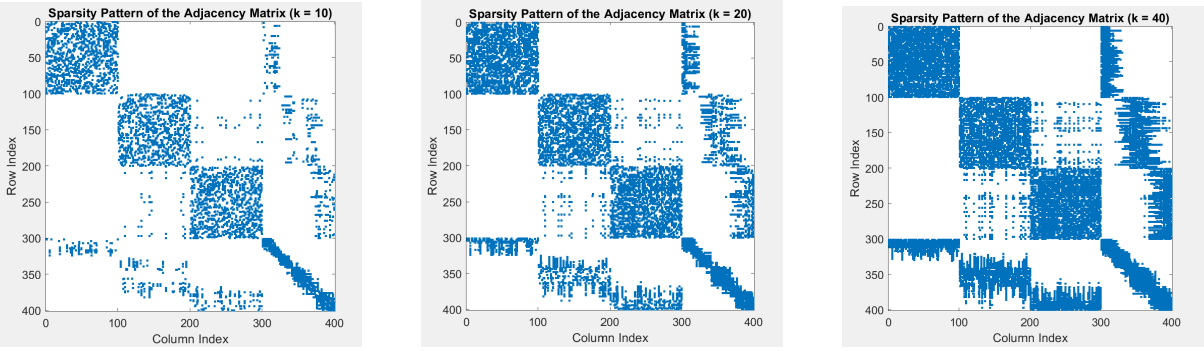
(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 1: Visualization of adjacency matrices W for the Circle dataset across different values of k.



(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 2: Visualization of adjacency matrices W for the Spiral dataset across different values of k.



(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 3: Visualization of adjacency matrices W for the Sphere dataset across different values of k.

Matrix Construction

To construct the required matrices for spectral clustering, the following steps were followed:

- **Computing the Degree Matrix D:** The degree matrix D is a diagonal matrix whose diagonal element D_{ii} means the sum of weights of edges attached to the node i. Mathematically:

$$D_{ii} = \sum_{j=1}^{n} W_{ij}, \quad \text{where } W_{ij} \text{ is the weight between nodes } i \text{ and } j.$$

  Since it is a diagonal matrix, the off-diagonal entries of D are zero. This denotes the degree of influence of a node in the similarity graph.

- **Constructing the Laplacian Matrix L:** Once D is computed, the unnormalized Laplacian matrix L is calculated as:

$$L = D - W,$$

  The adjacency matrix, where W is generated. The difference between the degree of nodes and the adjacency structure is captured in the Laplacian matrix, which forms the basis on which spectral graph analysis is performed.

- **Normalizing the Laplacian Matrix L_{norm}:** In order to improve robustness in clustering and make it perform well with graphs of different node degrees, the normalized Laplacian matrix L_{norm} has been formed. It is defined as follows:

$$L_{norm} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}},$$

  where $D^{-\frac{1}{2}}$ is the diagonal matrix with entries $1/\sqrt{D_{ii}}$. The normalization takes into account the changes in node degrees according to the variation of connectivity on the graph. The normalized Laplacian matrix is extremely popular in spectral clustering because it yields more trustworthy and user-friendly eigenvalues and eigenvectors.

Eigenvalue Computation

To analyze the graph structure and determine the number of connected components, the eigenvalues and eigenvectors of the Laplacian matrix are computed. The process involves the following steps:

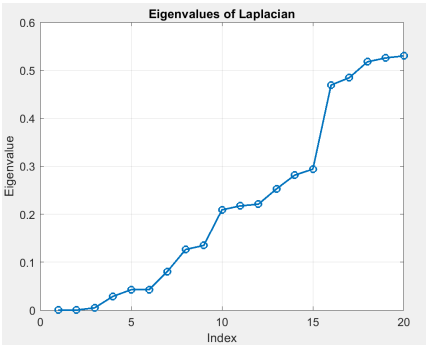
- **Compute the Eigenvalues of the Laplacian Matrices L and L_{norm}:** Both the unnormalized Laplacian matrix L and the normalized Laplacian matrix L_{norm} are derived from the degree matrix D and the adjacency matrix W. The eigenvalues of these matrices provide insights into the graph's structure. Specifically:

$$\text{Number of connected components} = \sum_{i} (\lambda_i \approx 0),$$

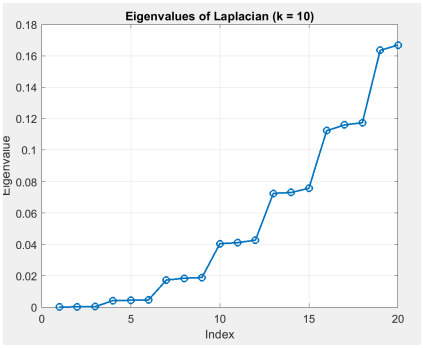
  where λ_i are the eigenvalues of L or L_{norm}. Each zero eigenvalue corresponds to a connected component in the graph. While L directly reflects the raw connectivity of the graph, L_{norm} adjusts for varying node degrees, making it more robust to changes in graph structure.

- **Determining the Number of Clusters (M):** Having determined the number of connected components, the number of clusters M is selected by examining the graph of eigenvalues. The eigenvalues closer to zero are chosen, as these correspond to the number of connected components in the graph.

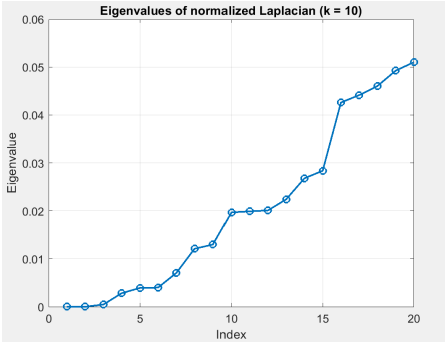
- **Eigenvalue Visualization:** The eigenvalues of both the unnormalized Laplacian matrix (L) and the normalized Laplacian matrix (L_{norm}) for the Circle, Spiral, and Sphere datasets were computed and plotted. The graphs below show the eigenvalues in ascending order, highlighting the small eigenvalues near zero that indicate connected components. These visualizations help in understanding the graph's structure and determining the appropriate number of clusters (M) based on the distribution of eigenvalues.



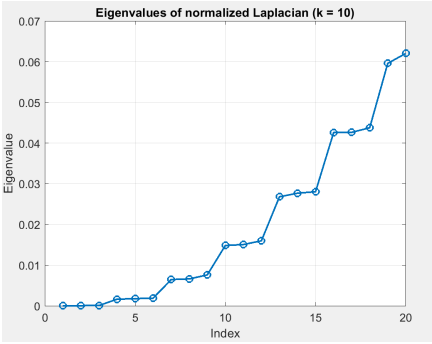
(a) L: Circle (k = 10).



(b) L: Spiral (k = 10).

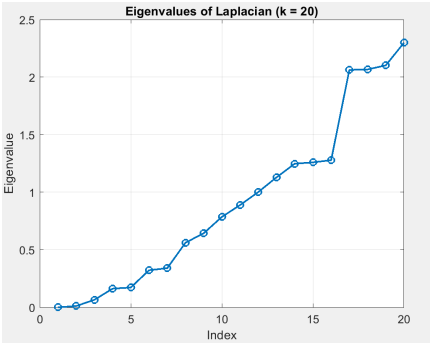


(c) L_{norm}: Circle (k = 10).

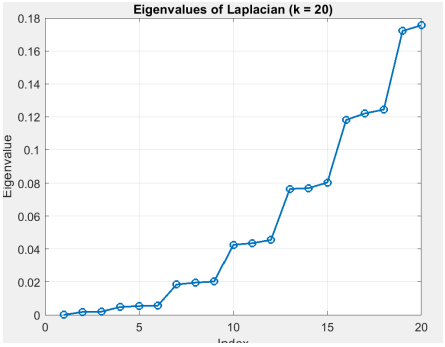


(d) L_{norm}: Spiral (k = 10).

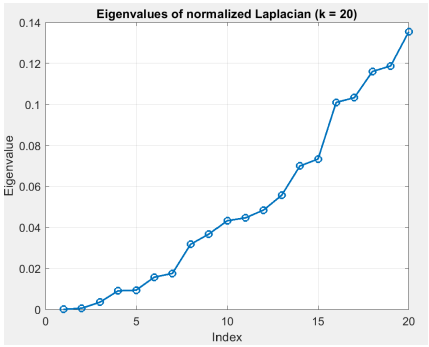
Figure 4: Eigenvalues of L and L_{norm} for Circle and Spiral datasets (k = 10).



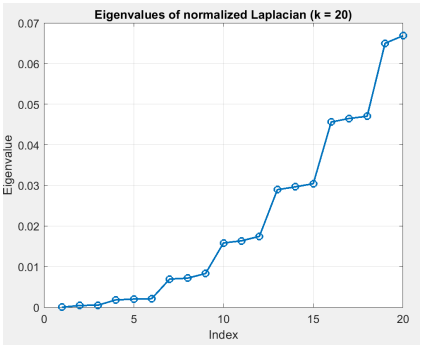
(a) L: Circle (k = 20).



(b) L: Spiral (k = 20).

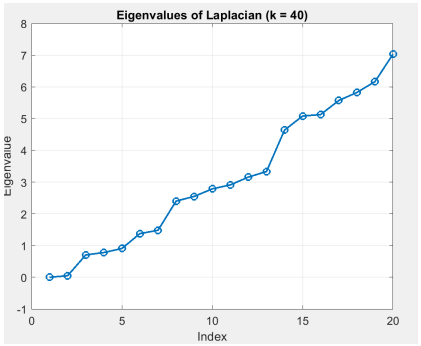


(c) L_{norm}: Circle (k = 20).

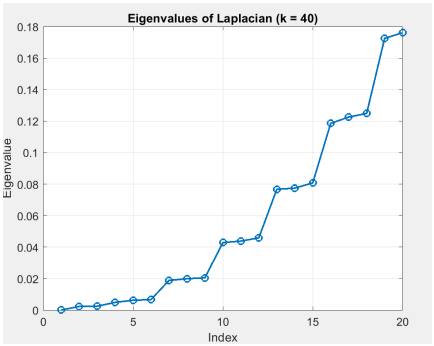


(d) L_{norm}: Spiral (k = 20).

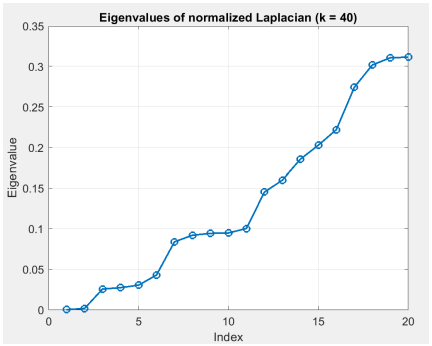
Figure 5: Eigenvalues of L and L_{norm} for Circle and Spiral datasets (k = 20).



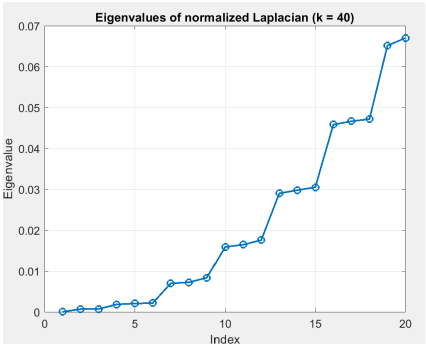
(a) L: Circle (k = 40).



(b) L: Spiral (k = 40).

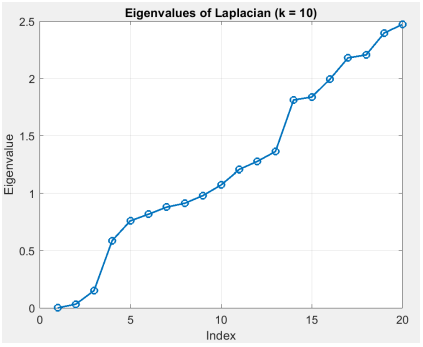


(c) L_{norm}: Circle (k = 40).

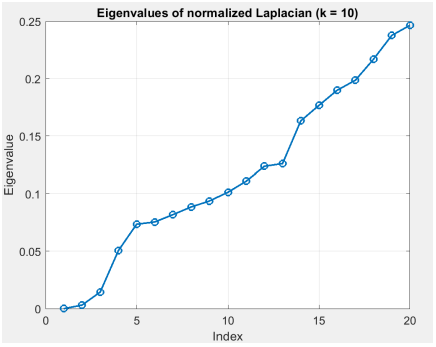


(d) L_{norm}: Spiral (k = 40).

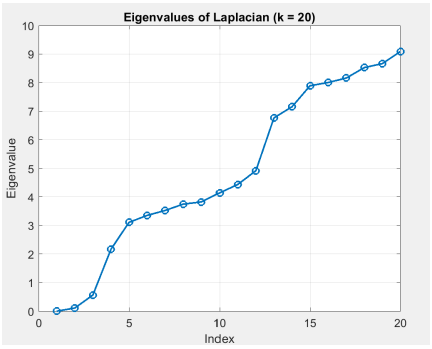
Figure 6: Eigenvalues of L and L_{norm} for Circle and Spiral datasets (k = 40).



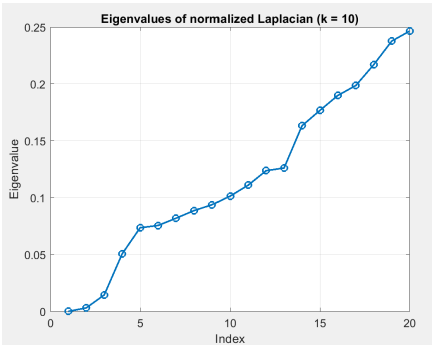
(a) L: Sphere (k = 10).



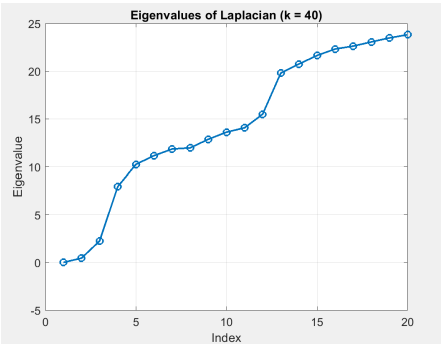
(b) L_{norm}: Sphere (k = 10).



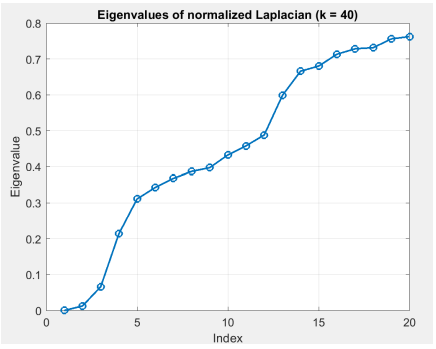
(c) L: Sphere (k = 20).



(d) L_{norm}: Sphere (k = 20).



(e) L: Sphere (k = 40).



(f) L_{norm}: Sphere (k = 40).

Figure 7: Eigenvalues of L and L_{norm} for Sphere dataset (k = 10, 20, 40).

Clustering

The clustering process is carried out based on the eigenvalues and eigenvectors of the Laplacian matrices (L and L_{norm}) and includes alternative clustering methods:

- **Determining the Number of Clusters:** The number of clusters (M) is determined by analyzing the eigenvalue plots. The number of eigenvalues close to zero corresponds to the number of connected components in the graph. This method applies to both the unnormalized Laplacian matrix (L) and the normalized Laplacian matrix (L_{norm}). For alternative clustering methods, the elbow method is also considered, where the point of maximum curvature in the variance plot is used to estimate M.

- **Clustering Methods:** - **Spectral Clustering:** After selecting M, the smallest M eigenvectors of the Laplacian matrix (L or L_{norm}) are extracted to form the

feature matrix $U \in \mathbb{R}^{N \times M}$. The rows of U are then clustered using the K-means algorithm. - **Hierarchical Clustering:** Hierarchical clustering does not require eigenvectors. It groups data points based on their similarity, forming a dendrogram. A cut-off level is chosen to define M clusters. - **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN identifies clusters based on density. It is particularly effective for detecting clusters of arbitrary shapes and separating noise points without needing the number of clusters M as input.

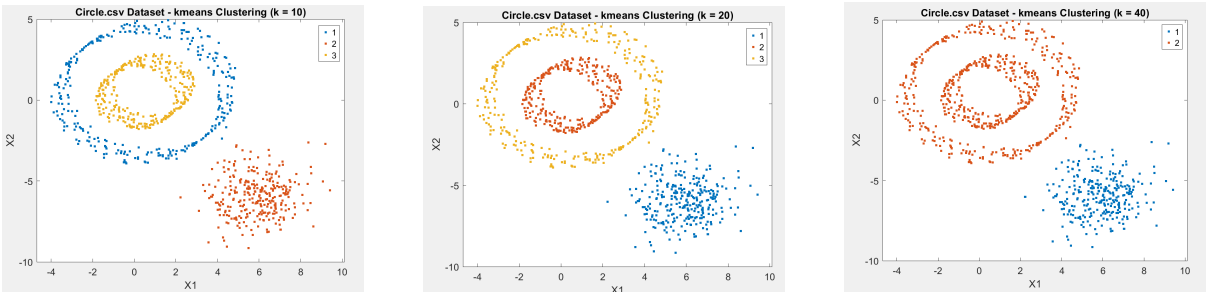
- **Mapping Clusters to Data Points:** For spectral clustering, the cluster assignments derived from K-means in the transformed feature space are mapped back to the original data points, assigning each point to a specific cluster. For hierarchical clustering and DBSCAN, the clusters are directly determined in the original space.

- **Visualization of Clusters:** The clustered data points are visualized for all methods for the unnormalized Laplacian matrix, we may get better and more efficient results with the normalized one in some cases. Each cluster is assigned a unique color, making it easy to compare the performance of spectral clustering, hierarchical clustering, and DBSCAN. Visualizations are provided for both the Circle and Spiral datasets.

3 Comparison of Clustering Methods for Circle Dataset

This section presents a comparative analysis of three clustering techniques—DBSCAN, Hierarchical Clustering, and K-means—applied to the Circle dataset. We evaluate their performance across different values of k and highlight their respective strengths and limitations.

3.1 K-means Clustering Results



(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 8: K-means clustering results for the Circle dataset.

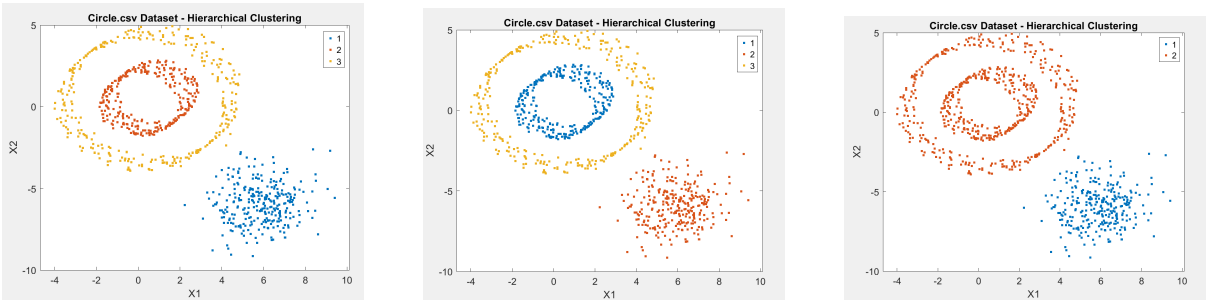
For $k = 10$ (Figure 8a) and $k = 20$ (Figure 8b), the number of clusters was selected as $M = 3$ based on the eigenvalue plot shown earlier (Figure 4c), which indicated three eigenvalues close to zero. This choice reflects the structure of the graph derived from the similarity matrix for k-nearest neighbors.

However, for $k = 20$, while the eigenvalue analysis suggested $M = 3$, reducing the number of clusters to $M = 2$ yields better results. With $M = 2$, the clustering aligns more closely with the natural structure of the concentric circles, resulting in a higher Silhouette

Score. This highlights that while eigenvalue-based selection of M is a good starting point, adjusting the number of clusters based on the dataset's geometric characteristics can lead to improved outcomes.

For $k = 40$ (Figure 8c), the clustering was performed with $M = 2$, which correctly separates the inner and outer circles. The results demonstrate that increasing k and reducing the number of clusters improves the alignment with the circular structure of the dataset.

3.2 Hierarchical Clustering Results



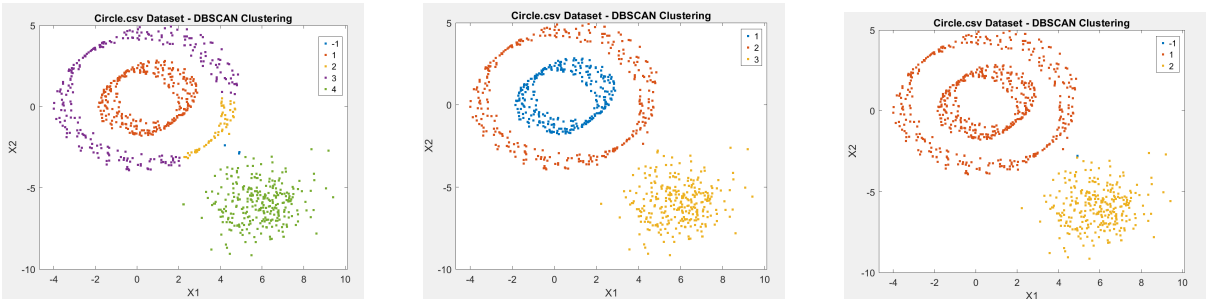
(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 9: Hierarchical clustering results for the Circle dataset.

For $k = 10$ (Figure 9a) and $k = 20$ (Figure 9b), hierarchical clustering separates the concentric circles but introduces additional boundaries that split the outer circle inappropriately.

At $k = 40$ (Figure 9c), the clustering more accurately captures the inner and outer circles, showing improved robustness with better connectivity.

3.3 DBSCAN Clustering Results



(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 10: DBSCAN clustering results for the Circle dataset. The clustering performance improves as k increases.

For $k = 10$ (Figure 10a), DBSCAN introduces noise points and over-segments the data into multiple clusters, particularly misclassifying points from the outer circle as noise or additional clusters. This is due to the smaller neighborhood size, which reduces the density required to form meaningful clusters.

For $k = 20$ (Figure 10b), the clustering significantly improves, clearly identifying the inner and outer circles with no noise points. The larger neighborhood size helps DBSCAN capture the structure more effectively, yielding clusters that align well with the concentric circles.

For $k = 40$ (Figure 10c), DBSCAN identifies the two concentric circles but introduces some noise points. The increased k provides greater connectivity, which helps form robust clusters, but the density variations in certain regions of the data lead to a small number of points being classified as noise.

3.4 Discussion

The results of the clustering methods on the Circle dataset reveal the following key insights:

- **DBSCAN:** This method is highly effective in identifying clusters with complex geometries, such as concentric circles. For smaller k-values, DBSCAN over-segments the data and misclassifies points as noise due to insufficient neighborhood size. However, as k increases, DBSCAN performs better, capturing the circular structure more accurately and handling noise effectively. Additionally, for higher values of M, such as $M = 6$, DBSCAN remains stable, consistently identifying the two concentric circles without introducing significant noise or additional clusters.

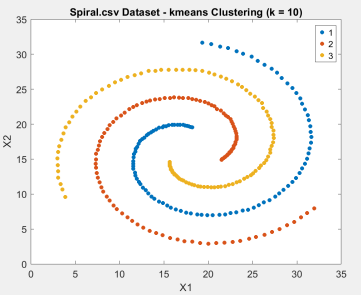
- **Hierarchical Clustering:** Hierarchical clustering is successful in capturing the circular structure of data and is less dependent on linear boundaries unlike K-means techniques. Smaller values of k introduce unnecessary boundaries around the outer circle and misclassify parts of it as well, although bringing in more number within k would facilitate a better separation of the two circles. Its reliance on the linkage method, coupled with its inability to handle noise as well as DBSCAN, still remain the limitations.

- **K-means:** With a smaller number of clusters, it turns out that the results on clustering do not coincide so well with the actual natural structure of the data. With an increase in k-value, separation is improved while K-means can manage to not completely cluster with the circular geometry. The main advantage of K-means is that it works well for optimizing compactness, which may lead to higher evaluation scores but does not necessarily reflect geometric accuracy.

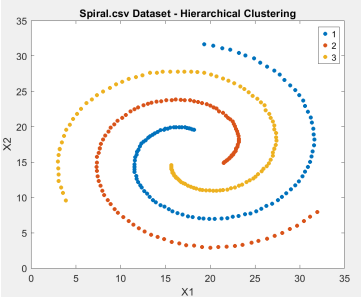
4 Comparison of Clustering Methods for Spiral Dataset

The clustering results for the Spiral dataset using DBSCAN, Hierarchical Clustering, and K-means suggest that there is little variation in resulting clusters among different choices of k when M is already fixed at 3. For example, for $k = 10$, 20, and 40, all three methods yield nearly identical clustering results in recognizing the spiral structure of the dataset with three clusters, regardless of the neighborhood size considered.

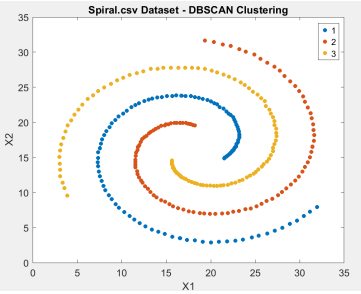
But if all clustering methods are also evaluated in terms of different numbers of clusters to get the better perspective of how such methods fit into the structure of data for the implicit patterns beneath the spiral shape, further tests are being visualized for different values of M.



(a) K-means.

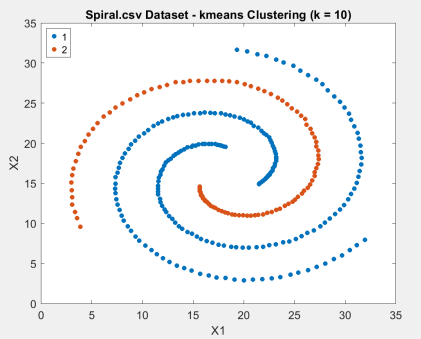


(b) Hierarchical.

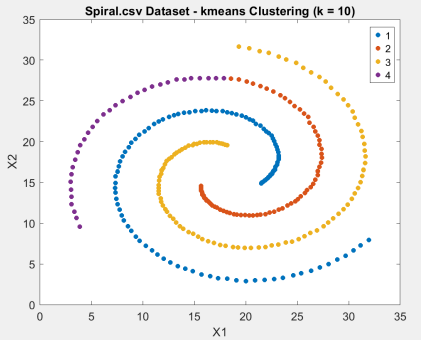


(c) DBSCAN.

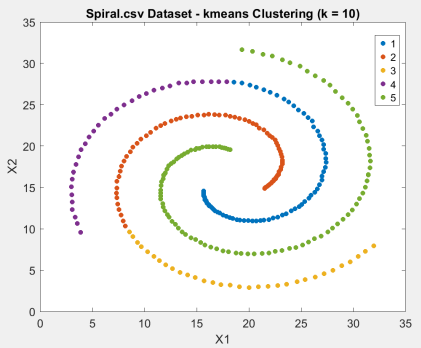
Figure 11: Clustering results for the Spiral dataset ($k = 10$) using K-means, Hierarchical Clustering, and DBSCAN.



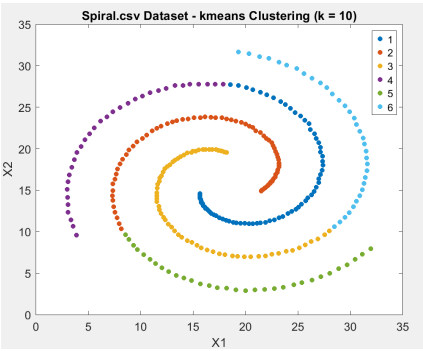
(a) $M = 2$



(b) $M = 4$



(c) $M = 5$



(d) $M = 6$

Figure 12: Clustering results for the Spiral dataset with different numbers of clusters (M).

The Spiral dataset clustering results will show that the interpretation of the data completely depends on the number of clusters (M). With M equal to 2, a bigger cluster simplifies the spirals, while with M equal to 4, 5, and 6 clustering the spirals is performed further and further into individual arms. It shows some flexibility in clustering methods meant for such complicated structures in data.

4.1 Discussion

The results of clustering spiral datasets using DBSCAN, Hierarchical Clustering, and K-mean are emphasized by the impact of the number of clusters (M) and the neighborhood

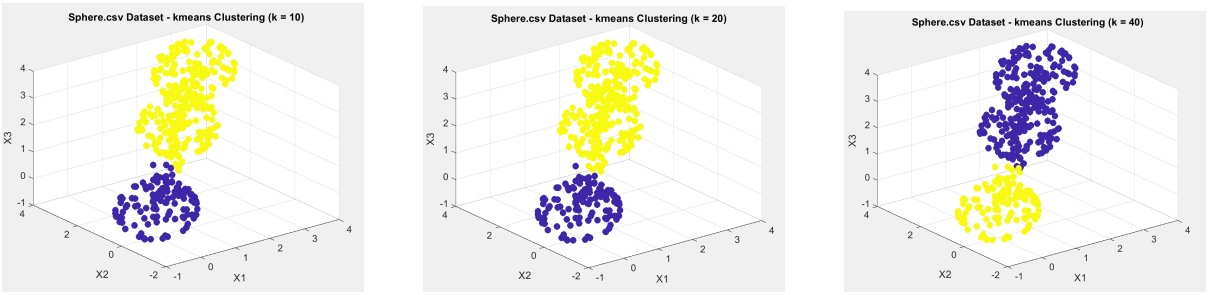
size (k), on the performance of every method:

- **DBSCAN:** DBSCAN effectively captures the spiral structure, particularly when the number of clusters is $M = 2$ or $M = 3$. For $M = 2$, the results are identical to $M = 3$, grouping the spirals into the same clusters without introducing additional noise. However, for $M = 4, 5$, and 6, DBSCAN identifies additional clusters while also labeling some points as outliers due to reduced density thresholds. This demonstrates DBSCAN's sensitivity to cluster numbers and its flexibility in detecting arbitrary shapes, albeit at the cost of increased noise for higher M.
- **Hierarchical Clustering:** Hierarchical Clustering produces results that are pretty similar to K-means when the clusters divide into finer sub-clusters (M¿3). Both methods parse through spirals making them exist in distinct regions but do so with certain boundaries drawn for the clusters. For M=2 or M=3, Hierarchical Clustering captures the general thrust of the spirals well. As M increases, so does the resemblance in clustering to K-means, cutting the spirals into smaller and smaller linear sections that do not necessarily follow the spiral geometry.
- **K-means:** K-means has the weakness of being very linear in nature. It does not capture the spiral structure-non-linearly. Thus, for $M = 2$, it will have the two big clusters of spirals but neglects the fine geometric ironies. As for $M > 3$, K-means will also give a partition of the spirals in increasingly smaller pieces similar to that in Hierarchical Clustering.

5 Comparison of Clustering Methods for Sphere Dataset

This section presents a comparative analysis of three clustering techniques—DBSCAN, Hierarchical Clustering, and K-means—applied to the Sphere dataset. The methods are evaluated across different values of k and their performance for capturing the spherical structure is discussed.

5.1 K-means Clustering Results



(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

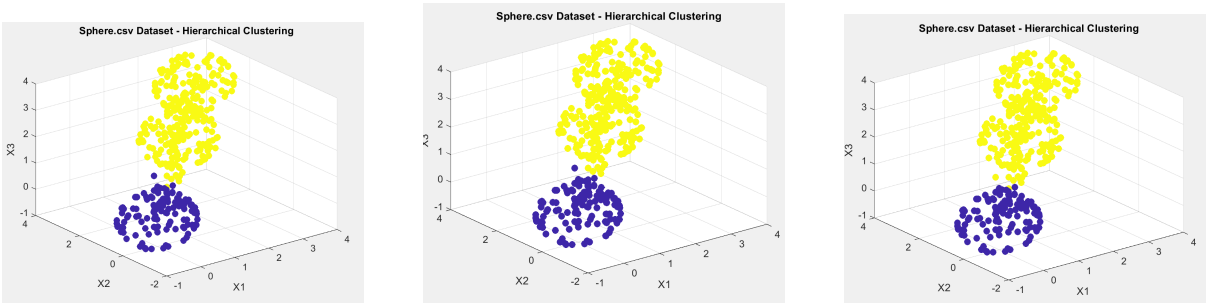
Figure 13: K-means clustering results for the Sphere dataset.

K-means Clustering Results

The Sphere data exhibits stable clustering results for all K-means clustering applications at k-values of $k = 10, 20, 40$. The method divides this data into two well-formed clusters

as indicated in the respective Figures: 13a, 13b and 13c.

5.2 Hierarchical Clustering Results

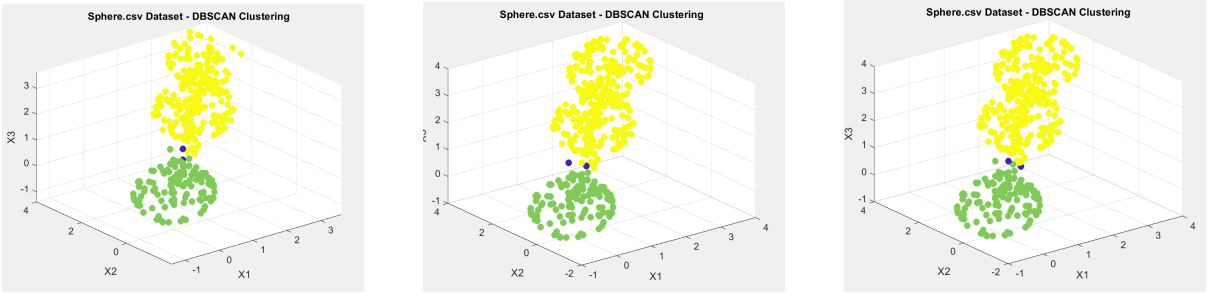


(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 14: Hierarchical clustering results for the Sphere dataset.

As illustrated in Figure 14, the hierarchical clustering is competent enough to capture the spherical structures of the data. The interpretation is also similar by having stable results at different k values. Note that Hierarchical Clustering yields results almost identical to K-means.

5.3 DBSCAN Clustering Results



(a) $k = 10$. (b) $k = 20$. (c) $k = 40$.

Figure 15: DBSCAN clustering results for the Sphere dataset.

For $k = 10$ (Figure 15a), DBSCAN separates the clusters effectively, introducing some noise points in low-density areas. At $k = 20$ (Figure 15b), the clustering improves, capturing the spherical structure more accurately. For $k = 40$ (Figure 15c), DBSCAN maintains consistent clustering quality, with the number of noise points remaining similar across all k-values.

5.4 Discussion

The results for the Sphere dataset using DBSCAN reveal the following characteristics:

- **DBSCAN:** From M=2 to M=3 it gives a clear separation of clusters well according to the characteristics of the data set. Increasing the value of M beyond M=3 has no

effect on the quality of results from DBSCAN; it has still been capable of grouping all spherical clusters accurately. For M=4: Grouping all spherical clusters accurately. But M=5 places nearly all points in one of the groups, labeling only a few points as noise. This shows DBSCAN is sensitive to hyperparameters such as (ϵ and $MinPts$) and shows a limitation of density clustering for very high M. However, DBSCAN carries the power of clustering spherical data, especially at lower M values.

6 Evaluation

The Silhouette Score and the Adjusted Rand Index (ARI) will be used as evaluation metrics for the clustering methods. While the Silhouette Score is apt for compact and well-separated clusters, it is obviously a good metric for the Circle and Sphere datasets. However, for the Spiral dataset, because the clusters are non-linearly structured and the Silhouette Score is not suited for non-Euclidean situations, its effectiveness is diminished. Therefore, ARI is used in the case of the Spiral since it assesses the degree of agreement between the clustering result and the ground truth.

Circle Dataset

For the Circle dataset, the Silhouette Score indicates the quality of the clustering results:

- For $k = 10$, K-means and Hierarchical Clustering achieve the highest Silhouette Score (0.377), outperforming DBSCAN, which scores 0.188. This demonstrates that while DBSCAN is effective in handling non-linear shapes, its performance is impacted by the smaller neighborhood size, which reduces cluster cohesion.
- For $k = 20$, all three methods—K-means, Hierarchical Clustering, and DBSCAN—achieve the same Silhouette Score (0.377). This indicates that, for this neighborhood size, the clustering results are similar across methods, with all forming well-defined clusters that align with the data's circular structure.
- For $k = 40$, K-means and Hierarchical Clustering achieve the highest Silhouette Score (0.758), reflecting their stability and ability to form well-defined clusters at larger k-values. In contrast, DBSCAN's score is lower (0.354), likely due to its sensitivity to density variations in the data, which results in less cohesive clusters.

Spiral Dataset

For the Spiral dataset, ARI is used to evaluate clustering performance due to its ability to handle complex, non-linear clusters:

- For $k = 10$, all three methods—K-means, DBSCAN, and Hierarchical Clustering—achieve perfect ARI scores (1.000), indicating ideal clustering. This reflects the ability of all methods to correctly identify the spiral clusters for this neighborhood size.
- For $k = 20$ and $k = 40$, all methods—K-means, DBSCAN, and Hierarchical Clustering—achieve perfect ARI scores (1.000), reflecting consistent performance and the ability to correctly identify the spiral clusters.

Sphere Dataset

For the Sphere dataset, the Silhouette Score is used to evaluate the clustering performance. The results show the following trends:

- For $k = 10$:
  - K-means achieves a Silhouette Score of 0.728, indicating well-formed clusters.
  - Hierarchical Clustering closely follows with a score of 0.721, demonstrating its ability to effectively capture the spherical structure.
  - DBSCAN achieves a slightly higher score of 0.724, showing its ability to handle the spherical structure at this neighborhood size.
- For $k = 20$:
  - Both K-means and Hierarchical Clustering perform equally well, achieving a Silhouette Score of 0.724, reflecting their stability in identifying the spherical clusters.
  - DBSCAN improves slightly with a score of 0.257, but it still lags behind due to its sensitivity to hyperparameters like ϵ and $MinPts$.
- For $k = 40$:
  - K-means achieves a Silhouette Score of 0.732, further reinforcing its effectiveness for compact, well-separated clusters.
  - Hierarchical Clustering slightly outperforms K-means with a score of 0.737, maintaining consistent clustering quality.
  - DBSCAN shows the lowest score (0.225), highlighting its challenges in handling densely connected clusters at higher k-values.

Key Functions

- `computeKNNGraph`: Constructs the k-nearest neighborhood graph.
- `computeLaplacians`: Computes W, D, L, and L_{norm}.
- `chooseClusters`: Determines M using the eigengap heuristic.
- `adjustedRandIndex`: Calculates ARI for clustering evaluation.

7 Conclusion

This report provided an in-depth study of spectral clustering and its comparative analysis with K-means, Hierarchical Clustering, and DBSCAN for three datasets: Circle, Spiral, and Sphere, bringing forth significant revelations concerning the merits and limitations of each method:

- **Spectral Clustering:** Spectral clustering is equipped to catch the intrinsic geometry of complex datasets such concentric circles, spirals and spheres. This technique utilizes the eigenvalues and eigenvectors of Laplacian matrices to perform robust clustering. The method of selecting the number of clusters M using the eigenvalue plot was shown to be critical for optimal results.

- **K-means:**The K-means clusters work fine in linear separability. Still, it cannot cluster non-linear and overlapping clusters well. K-means provides a little higher compactness scores in some instances; however, it cannot handle non-linear structures, and it is very limited when it comes to complex datasets like Spiral and Sphere.

- **Hierarchical Clustering:** Hierarchical clustering was found to provide fairly consistent performance across all datasets, capturing complex structures better even than K-means. Its reliance on linkage methods renders it flexible, but it has less ability to handle noise than DBSCAN.

- **DBSCAN:** When it comes down to datasets that are with arbitrary shapes and noise like that of the Spiral and Sphere datasets, DBSCAN shouts out as the best method of choice. However, it is very sensitive to hyperparameters which include ϵ and $MinPts$; hence a fine search for each user's requirement is needed.

- **Adjacency Matrix Construction:** The construction of similarity graphs using different k-values significantly impacts the clustering performance. Smaller k-values result in sparse adjacency matrices that may underrepresent connectivity, while larger k-values improve graph density but can overconnect distinct clusters.

Overall Findings: Spectral clustering consistently outperformed the other methods across all datasets, particularly for complex, non-linear structures. Among the other methods, DBSCAN performed best for non-linear clusters with noise, while Hierarchical clustering offered a balanced approach for interpretable and consistent results.