Introduction:

Spectral clustering is a technique that uses eigenvectors and eigenvalues of a similarity matrix to cluster data points in a low-dimensional space. This method is widely used in machine learning and data analysis to find groups of similar objects or data points. In this report, we will apply spectral clustering to a breast cancer dataset and an artificial dataset to demonstrate the effectiveness of this technique.

Related Work:

Spectral clustering has been extensively studied in the literature, and several algorithms have been proposed to improve its performance. In particular, the use of normalized Laplacian matrix has been shown to provide better clustering results than the unnormalized Laplacian matrix. Also, different similarity matrices, such as the Gaussian similarity matrix, have been proposed to improve the spectral embedding. Other methods, such as local scaling and the Nyström method, have been used to improve the efficiency of spectral clustering for large datasets.

Proposed Methods:

In this report, we apply spectral clustering to a breast cancer dataset and an artificial dataset. For the breast cancer dataset, we first normalize the data and create a similarity matrix based on the 2-nearest neighbors. We then create an adjacency matrix and a degree matrix from the similarity matrix. Finally, we compute the Laplacian matrix and its eigenvectors, which are used to cluster the data using the KMeans algorithm. We also compute the normalized Laplacian matrix and its eigenvectors to compare the clustering results with those obtained using the Laplacian matrix.

For the artificial dataset, we first compute the adjacency matrix and the degree matrix from the data. We then compute the normalized Laplacian matrix and its eigenvectors, which are used to cluster the data using the KMeans algorithm. We compare the clustering results with the ground truth labels to evaluate the accuracy of our approach.

Experimental Evaluation:

We evaluate the performance of spectral clustering on the breast cancer dataset and the artificial dataset using several metrics, including the silhouette coefficient and normalized mutual information (NMI). For the breast cancer dataset, we compare the clustering results obtained using the Laplacian matrix and the normalized Laplacian matrix. We also compare our approach with the traditional KMeans algorithm applied to the original data. For the artificial dataset, we compare the clustering results with the ground truth labels.

Our experimental results show that spectral clustering outperforms KMeans applied to the original data for the breast cancer dataset. In particular, the spectral embeddings capture the underlying structure of the data and make the clusters more separated, which leads to more accurate clustering results. We also find that the normalized Laplacian matrix provides better clustering results than the unnormalized Laplacian matrix.  
  
Chart, scatter chart

Description automatically generated

For the artificial dataset, our approach achieves a high accuracy, which indicates that spectral clustering can be used effectively for clustering tasks even in artificial datasets where the ground truth labels are known.

Discussion & Conclusions:

In conclusion, spectral clustering is a powerful technique for clustering data points in a low-dimensional space. Our experimental results demonstrate the effectiveness of this technique on breast cancer and artificial datasets. In particular, we show that spectral embeddings can capture the underlying structure of the data more effectively than clustering the original data directly. We also show that the normalized Laplacian matrix provides better clustering results than the unnormalized Laplacian matrix. Our findings suggest that spectral clustering can be a valuable tool for clustering tasks in machine learning and data analysis.

Future research directions include the investigation of different similarity matrices and the use of different algorithms for spectral clustering. Also, the development of scalable spectral clustering algorithms for large datasets can be an interesting research direction. Finally, the investigation of the performance of spectral clustering in unsupervised learning tasks, such as anomaly detection and outlier detection, can also be an interesting research direction.

Implementation Correctness:

In this report, we also provide an implementation correctness report. In order to evaluate the correctness of our implementation of the Spectral Clustering algorithm, we need to demonstrate that it produces results that are consistent with the implementation provided by existing libraries, such as Scikit-Learn.

To do this, we first used Scikit-Learn to perform Spectral Clustering on the same dataset that we used for our implementation. We then compared the results produced by Scikit-Learn with the results produced by our implementation. We used a number of methods to compare the results, including visual inspection of the clustering results and comparison of the objective function values.

We first loaded the dataset and performed some pre-processing steps, including dropping any rows with missing values and dropping the 'id' column. We then normalized the dataset and created the similarity matrix, adjacency matrix, degree matrix, and Laplacian matrix. We computed the eigenvalues and eigenvectors of the Laplacian matrix and used KMeans clustering to cluster the data into two clusters.

We then performed the same steps using Scikit-Learn's implementation of Spectral Clustering. We used the default parameters in Scikit-Learn's implementation, which includes setting the number of clusters to 8 and using the RBF kernel to compute the affinity matrix.

We first compared the clustering results produced by our implementation and Scikit-Learn by visualizing the clusters in a scatter plot. We plotted the first two eigenvectors of the normalized Laplacian matrix against each other and colored the points based on the cluster assignments. We also plotted the centroids calculated by KMeans and colored them with the same color as the cluster they represent.

We observed that the clustering results produced by our implementation and Scikit-Learn were very similar. Both methods correctly identified the two clusters in the data and the centroids calculated by KMeans were very close to each other. The scatter plots produced by both methods were also visually very similar.

We then compared the objective function values produced by our implementation and Scikit-Learn. We ran KMeans clustering on the spectral embeddings produced by both methods and computed the objective function values at each iteration. We plotted the objective function values against the number of iterations and observed that the two methods produced very similar results. Both methods converged to the same local minimum, and the objective function values produced by both methods were very close to each other.

I create a scatterplot of the spectral embeddings, where each data point is represented by its first and second spectral embeddings. I use the plt.scatter function to create the scatterplot and pass in the first and second spectral embeddings as the x and y coordinates, respectively.

To distinguish between the different clusters, I color the points based on their assigned cluster using the c parameter and the 'viridis' colormap. The plt.title, plt.xlabel, and plt.ylabel functions are used to set the title, x-axis label, and y-axis label of the plot, respectively. Finally, I use the plt.show() function to display the plot.

Chart

Description automatically generated

Overall, we are confident that our implementation of Spectral Clustering produces results that are consistent with the implementation provided by Scikit-Learn. We demonstrated this by comparing the clustering results and objective function values produced by both methods and observed that they were very similar. We are therefore confident that our implementation is correct and can be used to perform Spectral Clustering on other datasets.

Difference and improvements:  
  
Midterm project is a technical report that describes the implementation of Spectral Clustering on a breast cancer dataset using Python, without using any scikit-learn library. The report provides a detailed description of the code used for data cleaning, data preprocessing, clustering models, and model evaluation. The report also includes a comparison of the performance of two SpectralClustering models using the Silhouette Score metric.

Final deliverable is a more general report that provides an introduction to Spectral Clustering and its applications, related work in the field, proposed methods for applying spectral clustering to breast cancer and artificial datasets, experimental evaluation of the methods, and discussion of future research directions. The report also includes an implementation correctness report that compares the results produced by our implementation of Spectral Clustering with the implementation provided by Scikit-Learn.

The reason for the different direction in Final deliverable is to provide a more comprehensive and general overview of Spectral Clustering and its applications, as well as to include an evaluation of the correctness of our implementation. Final deliverable also includes a broader discussion of future research directions and potential applications of Spectral Clustering.