Practical 2 - Hierarchical Clustering and Linkage Measures

Introduction to Machine Learning
Laura Lall, Rachel Yang

I. Introduction

This report presents the implementation and analysis of agglomerative hierarchical clustering on a dataset containing 200 two-dimensional feature vectors. The clustering is performed using different linkage measures: single, average, complete, and Ward linkage. The number of clusters (K) is varied among 2, 3, and 4 to observe clustering behavior and evaluate performance using the silhouette score.

II. Methods

Agglomerative hierarchical clustering is a bottom-up clustering approach where each data point starts as its own cluster, and clusters are iteratively merged based on a predefined linkage criterion until a stopping condition is met.

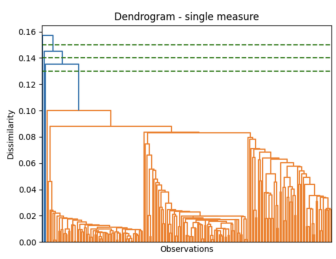
Single Linkage: merges clusters based on the minimum distance between any two points in the clusters.
Average Linkage: uses the average pairwise distance between points in different clusters.
Complete Linkage: merges clusters based on the maximum pairwise distance between points in different clusters.
Ward Linkage: minimizes the increase in variance when clusters are merged, making it effective in minimizing intra-cluster variance.

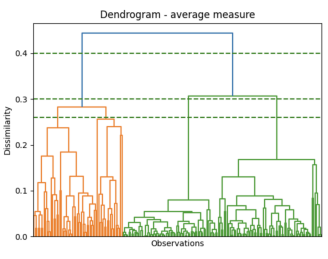
Implementation methods

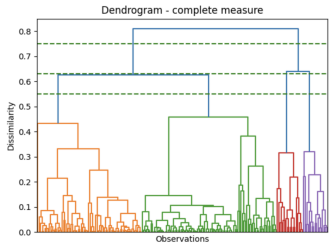
plot_dendrogram: this function generates the dendrogram for a given linkage measure and calculates the cut-off thresholds for different numbers of clusters.
agglomerative_clustering: this function performs the clustering for a given linkage measure and number of clusters, and it calculates the silhouette score.
read_data: this function reads the data from the CSV file.
plot_data_using_scatter_plot: this function plots the original data points.

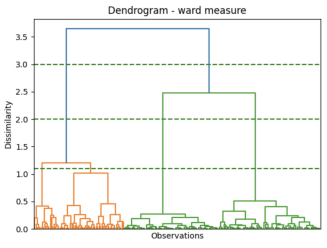
III. Results

Dendrograms were generated for each linkage measure, with cut-off lines indicating the chosen cluster thresholds (K = 2, 3, 4).

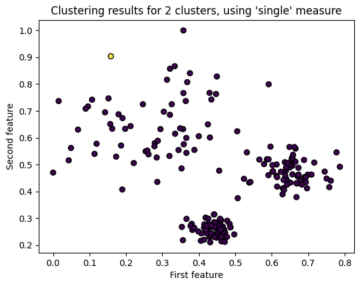


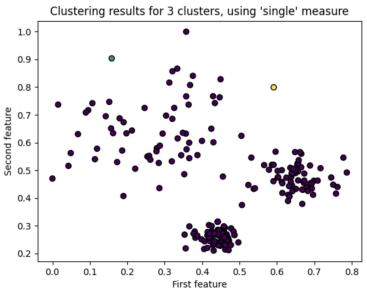


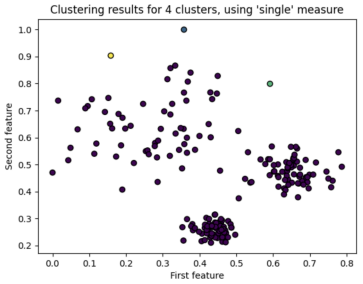


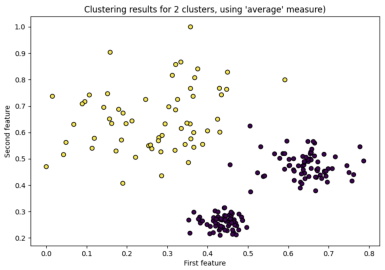


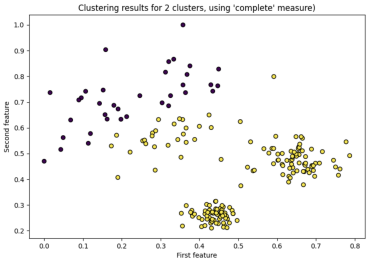
Scatter Plots include the scatter plots of the original data and the clustering results for each linkage measure and each value of K.

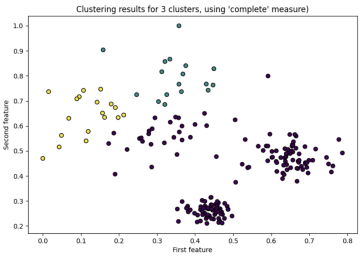


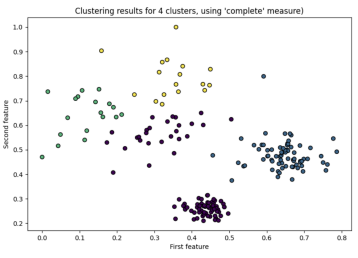


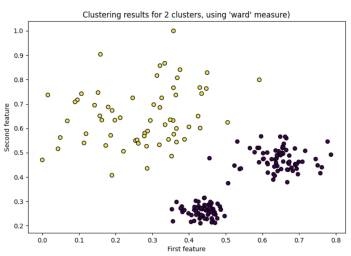


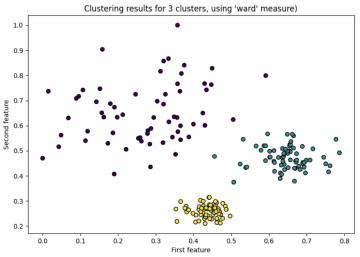


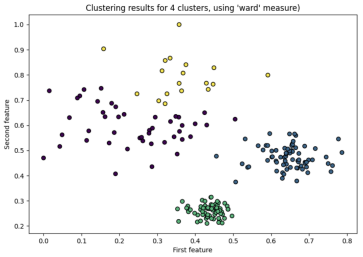












Linkage Method	**K=2**	**K=3**	**K=4**
Single	0.3773	0.1805	0.1852
Average	0.5412	0.6513	0.6269
Complete	0.4654	0.4207	0.4902
Ward	0.5412	0.6510	0.6251

IV. Discussion

Qualitative:

Number of Clusters

K=2: With only two clusters, the data is divided into two broad groups. This provides a preliminary separation of the data into distinct "piles," which can be useful for identifying high-level patterns. However, the simplicity of two clusters may not capture finer structures within the data.

K=3: Increasing the number of clusters to three allows for a more nuanced separation of the data. This often reveals intermediate groupings that are not apparent when K=2. For example, if the data has three natural subgroups, K=3 will better reflect this structure.

K=4: With four clusters, the data is divided into even finer groups. This can be particularly useful when the data has complex or overlapping structures. In many cases, K=4 provides the best separation, as it captures more subtle distinctions between data points.

Cluster Separation and Cohesion

As K increases, the clusters become more refined, but there is a trade-off between separation and cohesion. While K=4 may provide the best separation, it can also lead to smaller, less cohesive clusters, especially if the data does not naturally divide into four distinct groups. K=2 and K=3 often produce more cohesive clusters, but they may not capture all the underlying patterns in the data.

Dendrogram Interpretation

Single linkage: The clustering structure is unbalanced with many small clusters at the lower levels. Large gaps between earlier merges indicate that the clusters are not separated well and there is a clear chaining effect, meaning that the clusters are poorly defined.

Average linkage: The clusters are more balanced than single linkage and merges are evenly distributed across different levels. The cut-off thresholds indicate that there is decent clustering at k=3 and k=4.

Complete linkage: The clusters are well-separated, since the chaining effect that happens with single linkage was avoided. There are large dissimilarity jumps at the higher leels, meaning that there is a stronger separation between clusters.

Ward linkage: There is clear separation between the clusters and intra-cluster variance is minimized, making it the most balanced clustering from the 4 different methods.

Quantitative:

Single Linkage: the silhouette scores for single linkage were consistently lower compared to the other linkage measures, especially for k = 3 and k = 4. This indicates that single linkage tends to produce less cohesive and less separated clusters, particularly for larger k values.
Average Linkage: average linkage consistently produced the highest silhouette scores across all k values, with the highest score of 0.6513 for k=3. This suggests that average linkage is effective in producing well-separated and cohesive clusters.
Complete Linkage: the silhouette scores for complete linkage were moderate, with the highest score of 0.4902 for k=4. While complete linkage produced compact clusters, the separation between clusters was not as strong as with average linkage.
Ward Linkage: ward linkage performed similarly to average linkage, with high silhouette scores for k=3 (0.6510) and k=4 (0.6251). This indicates that Ward linkage is also effective in producing balanced and meaningful clusters.

Conclusion:

In conclusion, average and Ward linkage provided the best clustering results and highest silhouette scores. Complete linkage provided compact clusters, but the separation was not as strong as average or Ward linkage. We conclude that the optimal number of clusters was k =

3, since it resulted in high silhouette scores and well-separated clusters for both average and Ward linkage. While k = 4 divided the data into even finer groups, it had slightly lower silhouette scores, which indicates some over-segmentation.
The workload was divided efficiently, with Laura implementing most of the coding and Rachel writing most of the report and analyzing the results.