Lab Experiment-3

Cluster Analysis

CS3103: Machine Learning
Department of Computer Science and Engineering

August 13, 2025

Experiment Name

Implement Hierarchical Clustering for Cluster Analysis using a Custom Function.

Aim

To implement hierarchical agglomerative clustering from scratch using custom functions for single, complete, average, and centroid linkages; generate dendrograms/cut-based partitions, and compare how linkage choices affect cluster formation on a given dataset.

Dataset

Table 1: Sample Data for Clustering Analysis

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Platform / Tools Used

- Python 3.x

- Jupyter Notebook / Google Colab / VS Code
- Libraries: `numpy`, `pandas`, `matplotlib`, `scikit-learn`, `seaborn`

Introduction

Hierarchical Agglomerative Clustering (HAC) is a bottom-up clustering method in which each data point starts as its own cluster, and pairs of clusters are successively merged based on a defined distance metric, until all points belong to a single cluster or a predefined stopping condition is met. Unlike flat clustering methods such as k-means, HAC produces a **hierarchy of clusters**, typically visualized with a dendrogram. This hierarchical representation allows the user to explore multiple levels of granularity by applying **cut-based partitioning** at different heights in the dendrogram.

A key factor influencing HAC results is the **linkage criterion**, which defines how the distance $D(A, B)$ between two clusters A and B is computed:

1. **Single linkage** (minimum distance):

$$D_{\text{single}}(A, B) = \min_{x \in A,\, y \in B} d(x, y)$$

   where $d(x, y)$ is the distance between points x and y.

2. **Complete linkage** (maximum distance):

$$D_{\text{complete}}(A, B) = \max_{x \in A,\, y \in B} d(x, y)$$

3. **Average linkage** (mean pairwise distance):

$$D_{\text{average}}(A, B) = \frac{1}{|A|\,|B|} \sum_{x \in A} \sum_{y \in B} d(x, y)$$

4. **Centroid linkage** (distance between cluster centroids):

$$D_{\text{centroid}}(A, B) = d\left(\mu_A, \mu_B\right)$$

   where μ_A and μ_B are the centroids of clusters A and B, respectively.

Implementing HAC from scratch with custom functions for these linkage methods enables a deeper understanding of the clustering process and offers flexibility for tailored applications. In this study, we construct such implementations, generate dendrograms, extract

clusters through cut-based partitioning, and compare how different linkage strategies influence the resulting cluster structures on a given dataset. This comparative analysis provides insight into the trade-offs between linkage methods, supporting informed decision-making in practical clustering tasks.

Tasks -1

1. Visualize the data using a Scatter Plot.
2. Define your own function for each cluster analysis.
3. Show the distance matrix.
4. Show the distance between each cluster and the number of clusters after each iteration.
5. Plot Hierarchical Clustering (Dendrogram) for each linkage.
6. Compare your result with class sklearn.cluster.AgglomerativeClustering()

Tasks-2

Implement Agglomerative clustering on the Iris dataset using sklearn.cluster import AgglomerativeClustering and show the dendrogram

Expected Outcomes

1. Successful implementation of Hierarchical Agglomerative Clustering (HAC) from scratch for single, complete, average, and centroid linkages.
2. Generation of the initial and updated distance matrices after each merge step.
3. Iteration-wise log of merged clusters, merge distances, and remaining number of clusters.
4. Dendrograms for each linkage type and cut-based partitions to obtain flat clusters.
5. Comparative analysis showing how different linkage choices affect cluster formation on the given dataset.

Conclusion

In this experiment, Hierarchical Agglomerative Clustering (HAC) was successfully implemented from scratch using custom functions for single, complete, average, and centroid linkages. The approach allowed a detailed understanding of how inter-cluster distances are computed and updated during the agglomeration process. Visualization through dendrograms and cut-based partitions demonstrated that the choice of linkage criterion significantly affects cluster shapes, compactness, and merging patterns. Single linkage tended to form elongated chains, complete linkage produced compact and well-separated clusters, average linkage balanced both behaviors, and centroid linkage grouped based on mean positions with occasional inversions. Overall, the study highlights that selecting an appropriate linkage method should be guided by the dataset's structure, the desired cluster properties, and the specific application context.

References

- https://www.geeksforgeeks.org/machine-learning/hierarchical-clustering/
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClus
  html
- https://builtin.com/machine-learning/agglomerative-clustering
- Python Documentation - https://www.w3schools.com/python/python_ml_getting_
  started.asp