Hierarchical Clustering: A Machine Learning Tutorial

Student ID : 23069222

Student Name: Shiva Sai Mallavarapu

GitHub Link: https://github.com/Shiv-web-lab/ML-Assignment.git

1. Introduction

Hierarchical Clustering is a powerful unsupervised learning technique for grouping similar data points. Unlike in K-Means, where one must predetermine the number of clusters, hierarchical clustering generates a nested tree (dendrogram) that illustrates cluster relationships at different levels. It is therefore a useful technique for exploratory data analysis as well as for visualizing data patterns.

There are two main approaches:

- **Agglomerative Hierarchical Clustering (AHC**) : A bottom-up approach where each data point starts in its own cluster, and recursively clusters are merged based on similarity.

- **Divisive Hierarchical Clustering** : A top-down approach where all points start in a single cluster and are recursively split.

The algorithm uses distance metrics like Euclidean distance and linkage methods (single, complete, average) to determine how clusters should be merged or split. It has widespread application in bioinformatics (gene expression analysis), marketing (customer segmentation), and social network analysis.

This tutorial will explain the working of hierarchical clustering, key concepts, Python code, model evaluation techniques, advantages, applications, and techniques to enhance clustering accuracy.**2. How 2.Hierarchical Clustering Works**

Hierarchical Clustering approach systematically builds up clusters by the calculation of between-cluster or between-data-point distances. The most prominent steps performed in the algorithm are:

1**. Compute the Distance Matrix :** Approximate the pairwise distances between data points based on a measure like Euclidean, Manhattan, or cosine similarity.

2**. Form Initial Clusters :** Initialize each data point as one cluster.

3. **Merge Closest Clusters (Agglomerative Approach)** : The closest two clusters are merged iteratively by a chosen linkage method:

- **Single Linkage** : Uses the minimum distance between points in two clusters.

- **Complete Linkage** : Uses the maximum distance between points in two clusters.

- **Average Linkage**: Uses the average of the distance between points within two clusters.

4. **Repeat Until One Cluster Remains :** The algorithm repeats until all points are combined into a single cluster, forming a hierarchical tree-like structure known as a dendrogram.

5. **Determine the Optimal Number of Clusters :** The dendrogram is studied to determine the best cut-off point for meaningful clusters, generally using methods like the elbow method or silhouette analysis.

Hierarchical clustering is a detailed representation of data relationships and is used heavily for applications like customer segmentation, taxonomy classification, and social network analysis.

3. Key Concepts

3.1 Distance Metrics

Distance metrics specify how similarity between data points is calculated:

- **Euclidean Distance:** Measures the straight-line distance between two points in space. It is the most popular metric.

$$d(A,B) = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (Ai - Bi)^2}$$

where A and B are two data points in an n-dimensional space.

- **Manhattan Distance:** Computes distance by adding the absolute differences along each dimension, good for grid-based data.

$$d(A,B) = \sum_{i=1}^{n} | Ai - Bi |$$

This is useful for data constrained to grid-like structures.

Cosine Similarity : Calculates the cosine of the angle between two vectors, usually used for text and high-dimensional data.

$$\cos(\theta) = A \cdot \frac{B}{\| A} \|\| B \|$$

where $A \cdot B$ is the dot product and ||A||, ||B|| are the magnitudes of vectors.

3.2 Linkage Criteria

Linkage criteria define how to compute distances between clusters when merging:

- Single Linkage: Ties clusters together by computing the minimum distance between any two points in the clusters.

$$d(C1, C2) = min_{a \in C1, b \in C2}\ d(a,b)$$

- **Complete Linkage:** Uses the maximum distance between any two points in distinct clusters.

$$d(C1, C2) = max_{a \in C1, b \in C2}\ d(a,b)$$

- **Average Linkage:** Uses the average distance between all the points in two clusters.

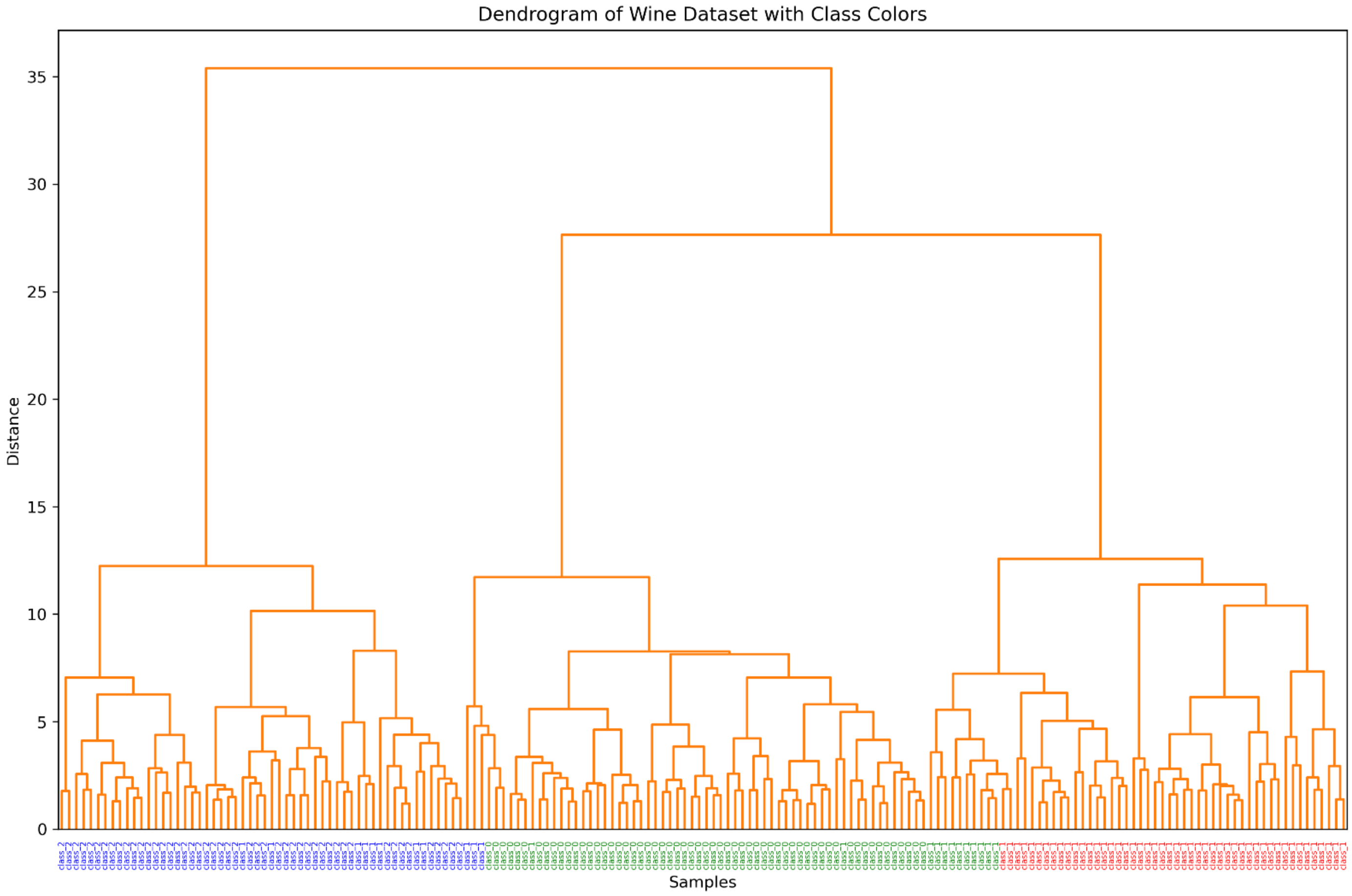
$$d(C1, C2) = \frac{1}{| C1} || C2 | \sum_{a \in C1} \sum\nolimits_{b \in C2} d(a,b)$$

- **Ward's Method :** Minimizes the variance within clusters, forming dense, well-balanced clusters.

$$d(C1, C2) = \frac{| C1 || C2 |}{| C1 | + | C2 |} \| \bar{C}1 - \bar{C}2 \|^2$$

where $\bar{C}1\ and\ \bar{C}2$ are the centroids of clusters $C1$ and $C2$.

3.3 Dendrogram



A dendrogram is a tree diagram of the merging process in hierarchical clustering. The vertical axis is the distance at which clusters merge. A larger vertical gap in the dendrogram is an optimal number of clusters, which can be determined by cutting the tree at the largest horizontal distance without crossing merges.

4. Implementation in Python

The code starts by importing core libraries like numpy, pandas, matplotlib, and clustering algorithms from sklearn. The Wine dataset is loaded using `load_wine()`, and the features are standardized utilizing `StandardScaler()` to bring all features into a similar range. The target labels (classes) are stored in `y`, and the feature data is stored in `X`.

Hierarchical clustering is done with `linkage()` function of `scipy.cluster.hierarchy` using "ward" method, reducing cluster variance. The dendrogram plot is drawn with `dendrogram()` function, where sample labels are coloured according to class using a dictionary where class is being mapped to colour. The plot is made more attractive by rotating the x-axis labels for better readability and saved as a high-resolution PNG image.

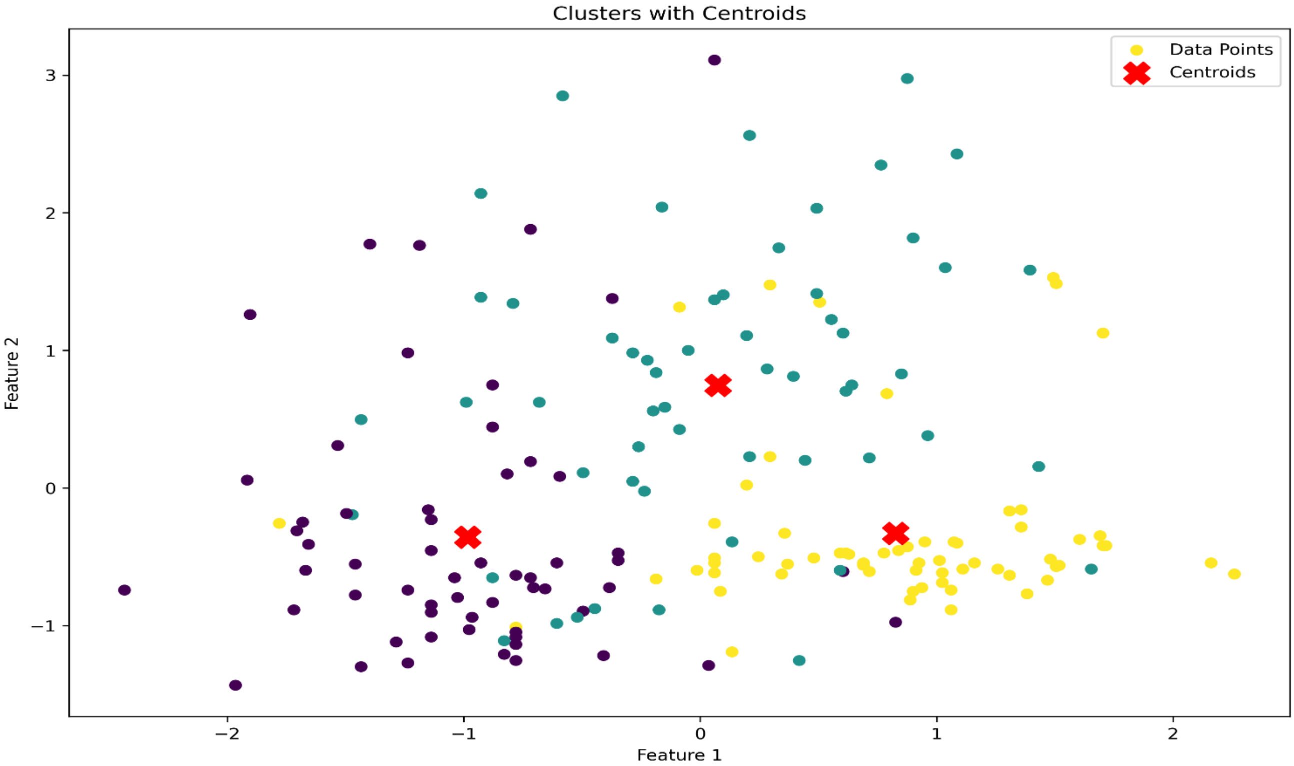
For performing clustering, `AgglomerativeClustering` is utilized to create three clusters. Centroids of every cluster are calculated as the average of points of every cluster. Scatter plot is plotted and centroids are marked in red colour. The scatter plot is also saved.

Lastly, the Adjusted Rand Index (ARI) is calculated by `adjusted_rand_score()`, which estimates the agreement between actual class labels and predicted clusters to provide a performance measurement for clustering.

5. Model Evaluation

The Adjusted Rand Index (ARI) of 0.7899 indicates superior performance for your model. ARI compares the clusters to actual labels, adjusting for chance. An ARI of 1 would represent a perfect clustering, and 0 would indicate random clustering, while negative numbers indicate worse-than-random clustering. An ARI of 0.7899, which is almost up to 1, indicates how well the model can cluster according to actual class labels.

The ARI value indicates that your clustering algorithm has identified the principal structure of the data with very high accuracy. This is a good result, particularly for unsupervised learning, in which there's no explicit supervision. It shows that your model is consistently finding clusters that are akin to the real categories of the dataset, but there are some misclassifications as well.



To continue refining the clustering:

- Experiment with different clustering algorithms like K-Means, DBSCAN, or Gaussian Mixture Models.

- Utilize dimensionality reduction techniques like PCA for maximizing the features and enhancing the clustering performance.

- Adjust the clustering model hyperparameters like the number of clusters or linkage measurement for closer proximity with actual labels.

It can be said that this ARI score measures great clustering precision, yet scope is always present for further refining and enhancement.

6. Advantages, Pros & Cons, and Comparison with Other ML Algorithms

Advantages:

No need to pre-specify the number of clusters: Hierarchical clustering, in contrast to K-Means, does not require the number of clusters to be prespecified.

Can capture nested clustering structures: Hierarchical clustering shows a tree-like clustering structure via the dendrogram, and cluster selection can be made flexibly.

Appropriate for small datasets with well-separated clusters: It is good if the dataset is small and the clusters are well separated.

Limitations:

Computationally expensive for large datasets: It has a time complexity of

$O(n2)$ which renders it impossible for very large datasets.

Sensitive to noise and outliers: The clustering structure can be distorted by noise data points.

Not scalable for high-dimensional data: The algorithm doesn't work well with datasets containing many features.

Comparison with K-Means

Feature	**Hierarchical Clustering**	**K-Means**
Cluster Structure	Tree-based (dendrogram)	Flat partitions
Scalability	Less scalable	More scalable
Handling Outliers	More sensitive	More robust
Number of Clusters	Not predefined	Predefined

Applications of Hierarchical Clustering

1. Bioinformatics

Hierarchical clustering finds widespread application in genetics for analysing relationship between different genes or species. It is used for building phylogenetic trees, for identifying patterns in gene expression, and in grouping similar sequences of DNA.

2. Marketing

Firms use hierarchical clustering to segment customers, grouping customers based on buying behavior, demographics, or interests. This enables focused marketing strategies, improving customer targeting and product recommendations.

3. Social Network Analysis

Hierarchical clustering helps identify clusters in social networks. It groups users based on interactions, common interests, or social relationships, which is useful in the identification of influencers or the setup of recommendation systems.

4. Anomaly Detection

Hierarchical clustering can be applied in finance and computer security to detect fraudulent transactions or network intrusions. Clustering normal transaction patterns will reveal outliers that correspond to fraud or cyber attacks.

Hierarchical clustering is applicable in many fields where inter-point relationships are to be determined at more than one level and thus can be a powerful exploratory data analysis tool.

8. How to Improve Accuracy

How to Enhance Accuracy of Hierarchical Clustering

1. Choosing the Correct Linkage Criteria

Various linkage techniques influence the formation of clusters. Elongated clusters are formed by single linkage, compact clusters are formed by complete linkage, and variance is minimized by Ward's method. The selection of the correct linkage depending on dataset features enhances the accuracy of clustering.

2. Preprocessing of Data

Hierarchical clustering is sensitive to noise and outliers. Removal of irrelevant features, handling of missing values, and numerical data normalization help generate more meaningful clusters. Standardization avoids letting variables of varying scales control the clustering process.

3. Choosing the Best Distance Measure

The choice of distance metric affects clustering performance. Euclidean distance is well-suited for continuous data, while cosine similarity is best for high-dimensional sparse data. Selection of a metric that best matches the nature of the data enhances accuracy.

4. Feature Selection and Dimensionality Reduction

Restricting the features reduces clustering efficiency and accuracy. Techniques like Principal Component Analysis (PCA) or domain-based feature selection reduce noise, rendering the clusters clearer and easier to understand.

By tuning these parameters with care, hierarchical clustering can yield improved accuracy and meaningful clusters, leading to data insights that are enhanced.

9. Conclusion

Hierarchical Clustering is a versatile unsupervised learning algorithm used to identify hidden patterns in data by organizing it in the form of a tree-like hierarchy. It does not require the number of clusters to be specified in advance, hence useful in exploratory data analysis. The algorithm is particularly valuable in the applications of bioinformatics, customer clustering, and outlier detection.

One of the strongest features of hierarchical clustering is its ability to capture nested cluster patterns, allowing for deeper understanding of data relationships. It is also constrained by being computationally costly and outlier and noisy data sensitive. Choosing the distance measure (e.g., Euclidean, Manhattan, or cosine similarity) and linkage style (e.g., single, complete, or Ward's) plays a crucial role in defining cluster boundaries and accuracy.

Despite its computational complexity, hierarchical clustering remains a valuable algorithm for small to medium-sized data. Proper data preprocessing, feature selection, and suitable choice of clustering parameters can significantly enhance its performance and accuracy.

10. References

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning.
- Scikit-learn Documentation: https://scikit-learn.org/stable/modules/clustering.html
- Tan, P., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining.