A Comprehensive Guide to K-Means and Hierarchical Clustering

Student Name: Syam krishna sujith Kolapalli

Student ID: 23017690

1. Introduction

What is Clustering?

Clustering is a very important unsupervised machine-learning technique that puts data into groups with similar inherent characteristics. Unlike in supervised learning, clustering does not require labeled data. Hence, it is very useful for exploratory data analysis (EDA). The ability to find hidden patterns without pre-labeled outcomes makes clustering a powerful tool in machine learning.

The main objective of clustering, therefore, is to assign all data points to clusters such that any two points in a cluster are more similar to each other than to points not in that cluster. In general, this is accomplished by some sort of similarity measure, usually a distance metric such as Euclidean distance. Because of this flexibility, clustering algorithms have proved their worth across several domains, including marketing (market segmentation), healthcare (grouping patients), and biology (classifying species or genetic data).

Applications of Clustering:

- **Market Segmentation:** Identifying distinct customer profiles based on purchasing behavior.
- **Image Segmentation:** Classifying different regions of an image based on pixel characteristics.
- **Anomaly Detection:** Identifying outliers or unusual patterns in financial transactions or network activity.
- **Biology and Genetics:** Grouping organisms or genes based on shared characteristics.

2. K-Means vs. Hierarchical Clustering

K-Means Clustering

K-Means is a partition-based clustering algorithm that divides data into a previously specified number of clusters, represented as (k). In this algorithm, each data point iteratively assigns to the closest cluster center, known as the centroid, and then it does an update of the centroids until convergence. This simplicity in approach has made K-Means highly efficient for use with large datasets.

However, K-Means also does not come without challenges. The main challenge is that it needs predefined number of clusters (k), which mostly is unknown. To address this, techniques like the "Elbow Method" or "Silhouette Analysis" are often used to determine the optimal k. Also, K-Means presupposes that clusters are spherical and of approximately equal size. This can lead to suboptimal clustering when data points are non-spherical or unevenly distributed.

Key Characteristics of K-Means:

- Efficient and scalable for large datasets.
- Requires specification of k clusters.
- Assumes clusters are spherical and of similar sizes.

Hierarchical Clustering

Hierarchical clustering, on the other hand, is a tree-based method that builds a hierarchy of clusters in a tree-like structure called a dendrogram. The hierarchy inherent in the cluster allows for different levels of granularity in the clustering, and it is easy to explore the data at higher or lower levels. Unlike K-Means, hierarchical clustering does not require pre-specifying the number of clusters.

There are two main types of hierarchical clustering:

i. **Agglomerative Clustering:** This is a bottom-up approach where each data point starts as its own cluster, and clusters are merged iteratively based on a chosen distance measure (such as single linkage or complete linkage).
ii. **Divisive Clustering:** This is a top-down approach, where the entire dataset starts as one cluster and is split iteratively.

While hierarchical clustering can produce more meaningful results, especially with irregularly shaped data, it is computationally expensive, particularly for large datasets. The time complexity of hierarchical clustering is $O(n^2)$ due to the need to compute pairwise distances between all data points.

Key Characteristics of Hierarchical Clustering:

- Does not require pre-specifying the number of clusters.
- Provides a hierarchical structure of clusters.
- Computationally expensive for large datasets.

Comparison between K-Means and Hierarchical Clustering:

- ✓ K-Means is fast and scalable, making it a good choice for large datasets where the number of clusters is known.
- ✓ Hierarchical Clustering provides a more flexible and interpretable structure, which is useful for smaller datasets or when exploring data at different levels of granularity.

3. Why Use the Iris Dataset?

Iris Dataset is one of the classic datasets in the machine learning community and is commonly used in teaching clustering and classification algorithms. It includes 150 samples of iris flowers-three species: Setosa, Versicolor, and Virginica where each sample is described by four numeric features:

- Sepal Length
- Sepal Width
- Petal Length
- Petal Width

The Iris dataset is ideal for clustering experiments for several reasons:

i. **Small Size:** With only 150 data points, it is computationally manageable for demonstrating clustering algorithms.
ii. **Clear Separability:** While there is some overlap between species (especially Versicolor and Virginica), the dataset's features generally allow for clear separability between the species, making it an excellent choice for clustering demonstrations.

iii. **Visualization-Friendly:** The dataset's four features allow for easy dimensionality reduction (e.g., using PCA or t-SNE) to visualize clusters in 2D or 3D plots.

By using the Iris dataset in this tutorial, we can demonstrate how different clustering algorithms behave and compare their performance in terms of cluster quality and interpretability.

4. Data Preprocessing

Clustering algorithm applications have to be prepared for data preprocessing before applying these to the Iris dataset. Preprocessing of the dataset cleans it, hence standardizing it and getting the dataset ready for clustering. The main steps involved in general during preprocessing are:

a. **Loading the Dataset**
We will use the Iris dataset from sklearn.datasets to load the data into our Python environment.

b. **Handling Missing Values**
The Iris dataset does not contain missing values, so no imputation or removal is necessary here. However, in other real-world datasets, missing data must be handled appropriately (e.g., using mean imputation or removing rows with missing values).

c. **Feature Scaling**
Clustering algorithms, especially K-Means, are sensitive to the scale of features. Therefore, we should standardize the features to have a mean of 0 and a standard deviation of 1. This is done using StandardScaler from sklearn.preprocessing.

d. **Dimensionality Reduction (Optional)**
While the Iris dataset has only four features, in some cases, dimensionality reduction techniques like PCA (Principal Component Analysis) or t-SNE (t-Distributed Stochastic Neighbor Embedding) can be used to reduce the dataset to two or three dimensions for easier visualization.

5. Applying Clustering Algorithms

Now that the dataset is preprocessed, we can apply K-Means and Hierarchical Clustering.

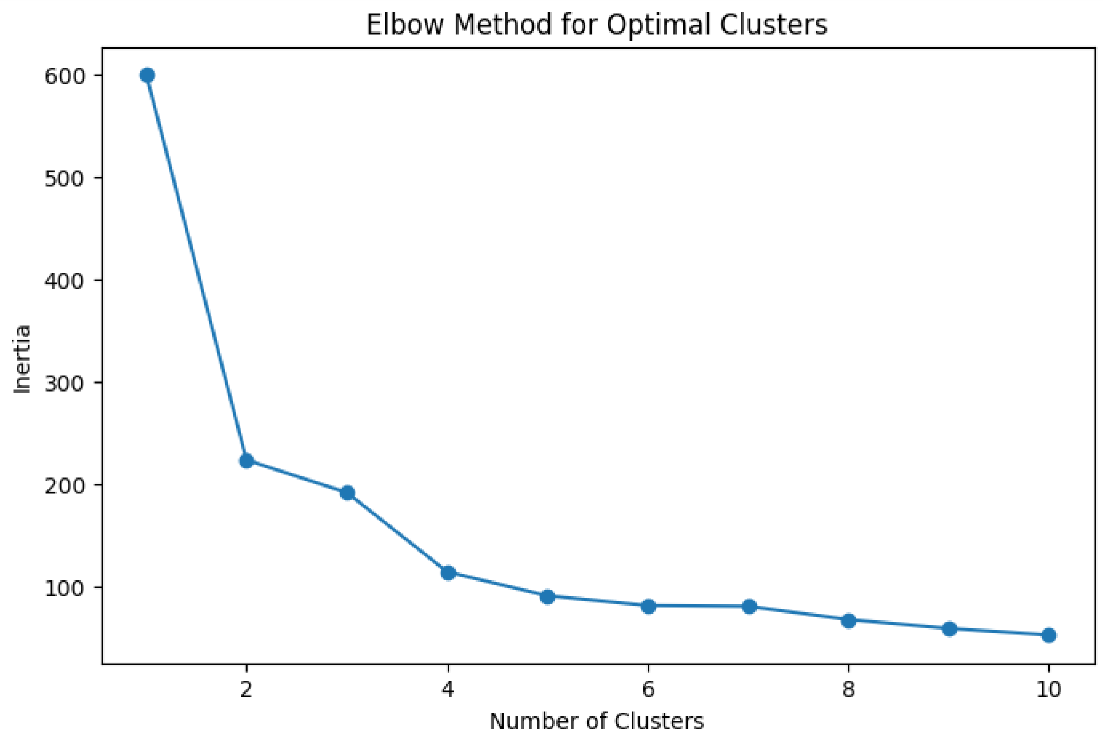
- **K-Means Clustering**
  We will use KMeans from sklearn.cluster to apply the K-Means algorithm. We will specify k=3 because we know there are three species in the dataset.
- **Hierarchical Clustering**
  We will apply Agglomerative Clustering, which is the most commonly used type of hierarchical clustering, using AgglomerativeClustering from sklearn.cluster.

6. Visualizing the Results

Following the clustering of data using K-Means and Hierarchical clustering, visual inspection of the cluster from both algorithms will be required to ascertain how each algorithm has fared in the grouping of the data. In this section, we shall visualize the resulting clusters by use of matplotlib and seaborn's scatter plots that will show points in a scatter plot classified under their respective cluster labels.

Determining the Optimal Number of Clusters Using the Elbow Method

To find out the optimal number of clusters for K-Means, we will use the Elbow Method. It works by doing K-Means clustering of data for a specified range of values of k and plotting the obtained inertia (within-cluster sum of squares). The "elbow" of the plot, where the inertia decreases at a slower rate, indicates the optimal number of clusters.



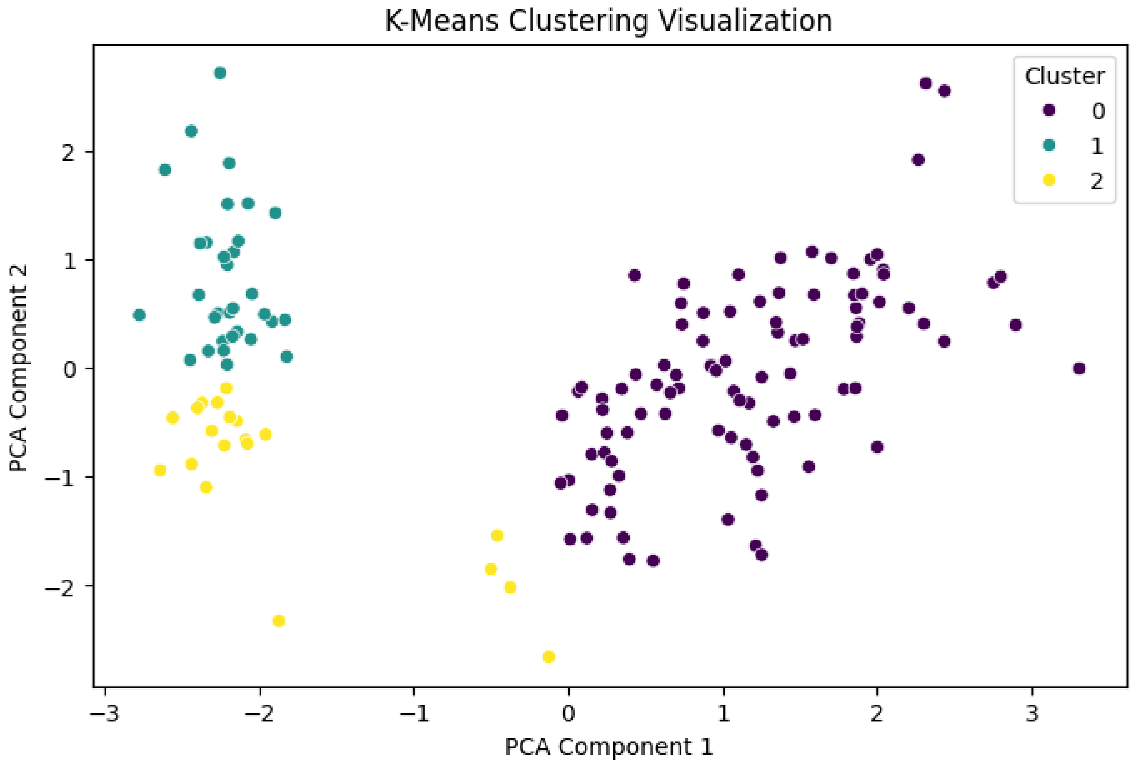
The curve tends to bend around 4 clusters, thus the best number of clusters. That would be an optimal balance between within-cluster variation minimization and maintaining reasonable cluster numbers.

K-Means Clustering and Visualization

For this tutorial, let us assume that 3 is the optimal number of clusters for this dataset. We will now fit the K-Means algorithm with cluster size 3 and visualize the results after reducing the dimensions of the data to two principal components by using Principal Component Analysis (PCA). This makes it easier to plot and visualize the clusters.

The visualization below shows three distinct clusters, aligning with the three iris species. This suggests the K-Means algorithm successfully separated the data based on the features used (likely sepal length, sepal width, petal length, and petal width).

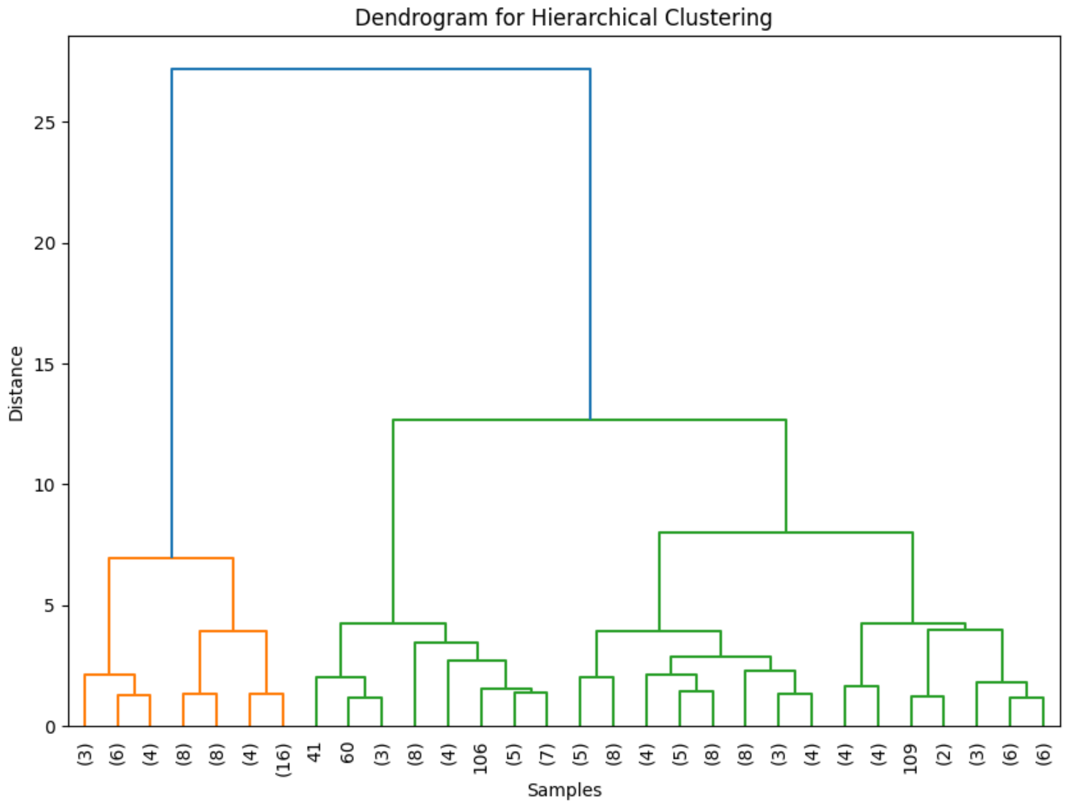
K-Means achieved a Silhouette Score of **0.479**, indicating better clustering performance on the Iris dataset. This suggests that K-Means grouped data points more cohesively within clusters and separated clusters better compared to Hierarchical Clustering.



Hierarchical Clustering and Visualization

Next, we apply Hierarchical Clustering using the Ward linkage method. This method minimizes the variance within each cluster during the hierarchical merging process.

The dendrogram visually represents the hierarchical relationships between the data points. The cut-off threshold (which defines the number of clusters) can be chosen based on the visual inspection of the tree.



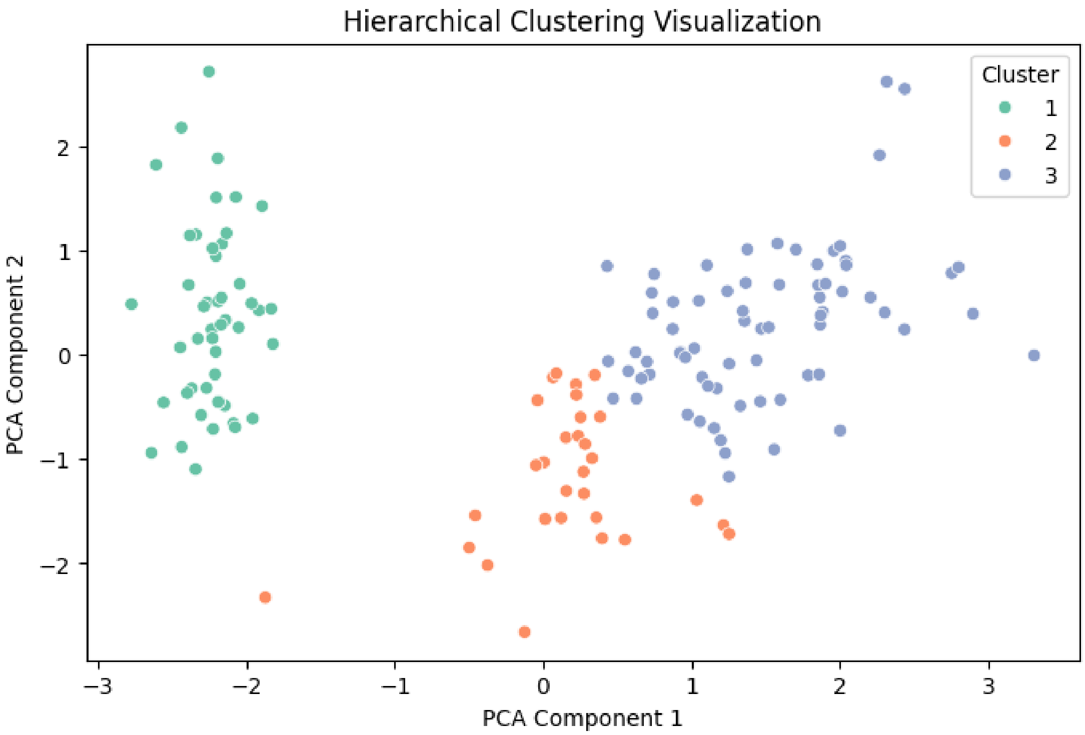
Assigning Clusters from Hierarchical Clustering

Once we have performed the hierarchical clustering, we use the fcluster function to assign the data points to clusters based on a pre-determined threshold (in this case, 3 clusters).

Visualizing Hierarchical Clustering Results

We can now visualize the results of the hierarchical clustering using the same PCA-based scatter plot as for K-Means clustering, allowing us to compare both clustering methods side-by-side.

This scatter plot shows how hierarchical clustering assigns data points to clusters, with different colors representing each cluster. Hierarchical Clustering achieved a Silhouette Score of 0.446, which is slightly lower than K-Means. This may indicate challenges in separating clusters optimally or merging some points prematurely.



Conclusion

In this tutorial, we demonstrated the fundamentals of K-Means and Hierarchical Clustering, focusing on their strengths, weaknesses, and practical applications. We used the Iris dataset to apply both algorithms, preprocess the data, and visualize the results. By the end of this guide, you should have a solid understanding of how to apply clustering algorithms to real-world datasets, how to interpret the results, and how to visualize clusters effectively. This tutorial provides a foundation for clustering techniques, which you can extend to more complex datasets and other machine learning tasks.

Reference

UCI Machine Learning Repository: Fisher, R. (1936). Iris [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C56C76.

Blog: Clustering with Confidence: A Practical Guide to Data Clustering in Python (https://medium.com/@nomannayeem/clustering-with-confidence-a-practical-guide-to-data-clustering-in-python-15d82d8a7bfb).

Key Scientific Papers:

1. "A Taxonomy of Machine Learning Clustering Algorithms, Challenges, and Future Realms"

This paper reviews traditional and modern clustering techniques, presenting a taxonomy and discussing challenges like scalability and dimensionality.

Reference: MDPI Applied Sciences 【100】 .

2. "Clustering Algorithms: A Comparative Approach"

This study compares the performance of clustering algorithms across diverse datasets, examining how parameter tuning and algorithmic choice influence outcomes.

Reference: PLOS ONE 【101】 .

3. "Big Data Clustering: Trends and Challenges"

Explores clustering techniques adapted for big data, addressing challenges in data dimensionality and distributed processing.

Reference: Check on scientific databases such as IEEE Xplore or Springer for similar works.