# Hierarchical Clustering

### I. Introduction

- **Definition:** A cluster analysis technique that builds a hierarchy of clusters, represented as a tree-like structure called a **dendrogram**. Unlike K-Means, it doesn't require specifying the number of clusters beforehand.
- **Goal:** To create a nested sequence of clusters, from individual data points to a single cluster containing all data points (or vice versa). This allows for exploration of data at different levels of granularity.
- **Key Idea:** Grouping or dividing clusters based on their similarity (or dissimilarity) in a hierarchical fashion.
- **Output:** A dendrogram, which visually illustrates the hierarchical relationships between data points and clusters. The height at which two clusters are merged (or a cluster is split) indicates their dissimilarity.
- **Applications:**
  - **Biology:** Phylogenetic analysis, gene expression studies.
  - **Marketing:** Customer segmentation based on behavior or demographics.
  - **Social Science:** Grouping individuals based on survey responses.
  - **Image Processing:** Image segmentation, object recognition.
  - **Document Clustering:** Organizing documents by topic.

### II. Types of Hierarchical Clustering

1. **Agglomerative (Bottom-Up):**
   - Starts with each data point as its own individual cluster.
   - Iteratively merges the closest pairs of clusters until a single cluster containing all data points is formed.
   - Also known as **AGNES** (Agglomerative Nesting).
   - More commonly used due to its conceptual simplicity and ease of implementation.
2. **Divisive (Top-Down):**
   - Starts with all data points in a single cluster.
   - Recursively splits the most heterogeneous cluster into smaller sub-clusters until each data point forms its own cluster.
   - Also known as **DIANA** (Divisive Analysis clustering).
   - Conceptually more complex and less commonly used in practice, especially for complete hierarchies. Can be more efficient if only a few top levels of the hierarchy are needed.

### III. The Agglomerative Clustering Process

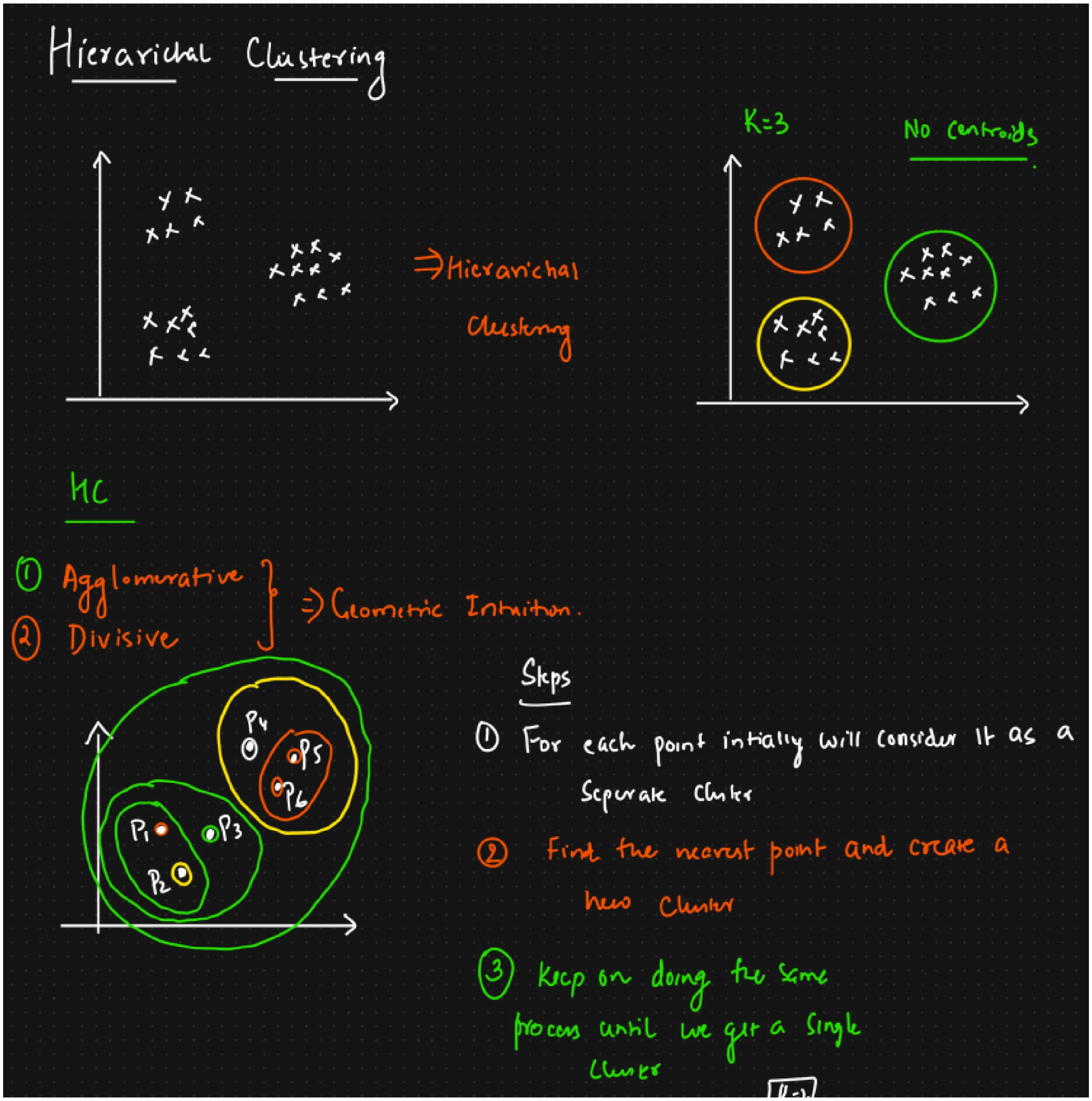
1. **Initialization:** Each data point is considered a single cluster.

2. **Compute Proximity Matrix:** Calculate the pairwise distances (dissimilarities) between all clusters. Common distance metrics include:
    - **Euclidean Distance:** Straight-line distance.
    - **Manhattan Distance:** Sum of absolute differences along each dimension.
    - **Cosine Similarity/Distance:** Measures the cosine of the angle between two vectors (similarity), or 1 - cosine similarity (distance). Useful for text and high-dimensional data.
3. **Merge Closest Clusters:** Find the two clusters with the minimum distance according to the chosen **linkage criterion** and merge them into a single new cluster.
4. **Update Proximity Matrix:** Recalculate the distances between the new cluster and all remaining clusters using the chosen linkage criterion.
5. **Repeat:** Steps 3 and 4 are repeated until all data points belong to a single cluster.

**IV. Linkage Criteria (Determining Distance Between Clusters)**

The choice of linkage criterion significantly affects the shape and characteristics of the resulting clusters. Common methods include:

- **Single Linkage (Nearest Neighbor):** The distance between two clusters is the minimum distance between any two points in the two clusters.
    - Tends to produce long, chain-like clusters.
    - Sensitive to noise and outliers.
- **Complete Linkage (Farthest Neighbor):** The distance between two clusters is the maximum distance between any two points in the two clusters.
    - Tends to produce more compact, spherical clusters.
    - Less prone to chaining but can split large clusters prematurely.
    - More sensitive to outliers.
- **Average Linkage (UPGMA - Unweighted Pair Group Method with Arithmetic Mean):** The distance between two clusters is the average of the distances between all pairs of points, one from each cluster.
    - A good compromise between single and complete linkage.
    - Less sensitive to outliers than single or complete linkage.
- **Centroid Linkage:** The distance between two clusters is the distance between their centroids (mean vectors).
    - Can sometimes lead to inversions in the dendrogram (non-monotonicity).
- **Ward's Method:** Merges the two clusters that result in the minimum increase in the total within-cluster variance (sum of squared distances to the cluster centroids).
    - Tends to produce compact, evenly sized clusters.
    - Often a good default choice when there's no strong theoretical justification for another method.

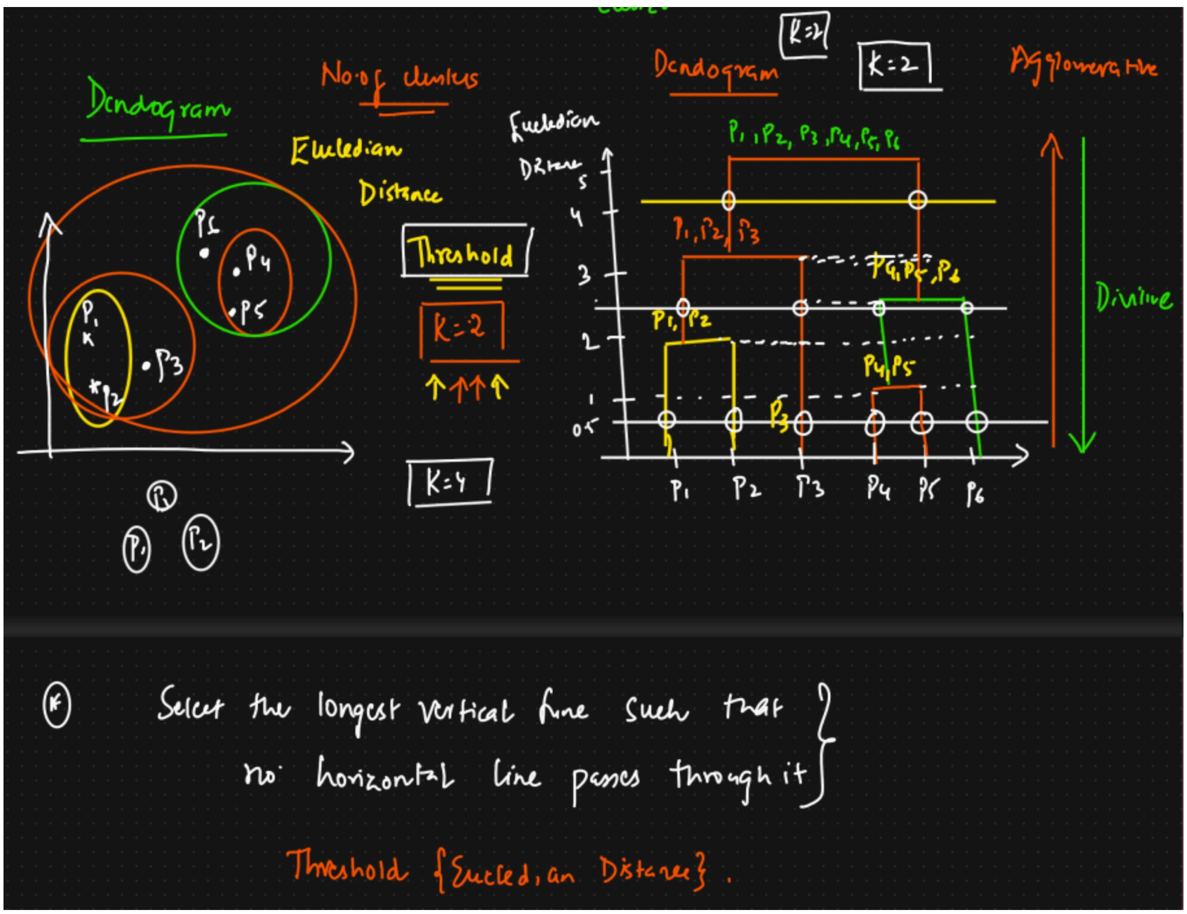


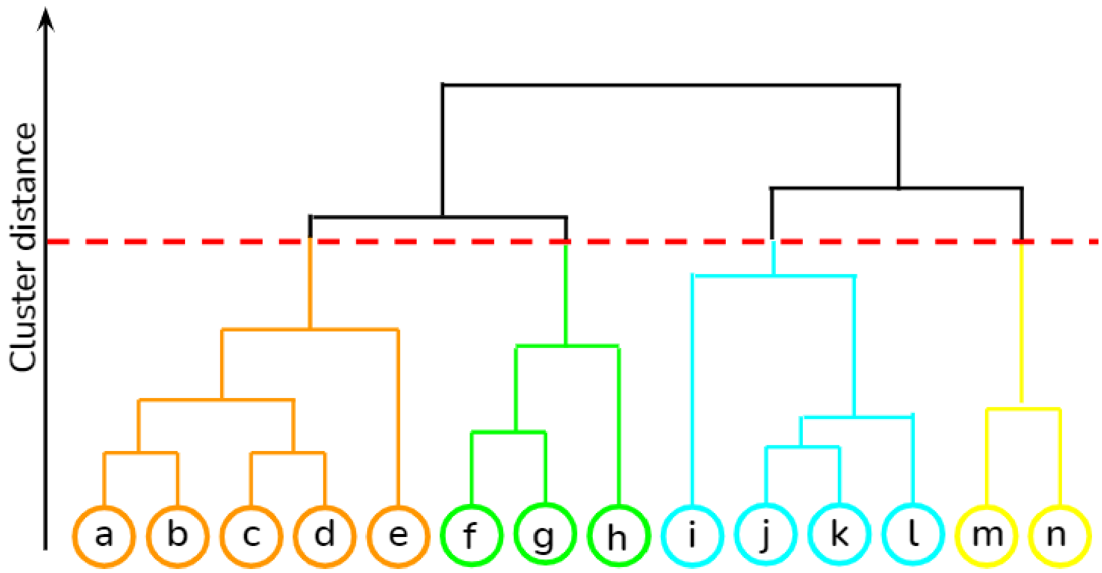

### V. The Divisive Clustering Process

1. **Initialization:** All data points are in one large cluster.
2. **Choose Cluster to Split:** Select the "most heterogeneous" cluster to split (e.g., the one with the largest diameter or variance).
3. **Split the Cluster:** Divide the chosen cluster into two or more sub-clusters using a flat clustering algorithm (like K-Means) or by finding the most dissimilar points within the cluster.
4. **Repeat:** Steps 2 and 3 are repeated recursively on the resulting sub-clusters until each data point is in its own cluster or a stopping criterion is met.

### VI. Interpreting the Dendrogram

- The dendrogram is a tree diagram that illustrates the merging (agglomerative) or splitting (divisive) process.
- **Leaves:** Represent the individual data points.
- **Nodes:** Represent the clusters formed at each step.
- **Height of Branches:** The vertical height at which two branches merge (or a branch splits) indicates the distance (dissimilarity) between the clusters at that point. Shorter heights indicate more similar clusters.
- **Determining the Number of Clusters:** By visually inspecting the dendrogram, you can choose a horizontal line that intersects the tallest vertical lines without crossing any clusters. The number of vertical lines intersected by this horizontal line represents a potential number of clusters.






**VII. Advantages of Hierarchical Clustering**

- **No need to pre-specify the number of clusters (k):** The dendrogram provides a full hierarchy, allowing you to choose the number of clusters after the analysis.
- **Provides a hierarchical structure:** Reveals nested relationships between clusters, offering more insight into the data's organization.
- **Easy to visualize results:** The dendrogram is an intuitive way to understand the clustering process.
- **Flexibility in choosing distance metrics and linkage criteria:** Allows adaptation to different data types and cluster characteristics.
- **Can be less sensitive to the initial conditions** compared to K-Means (especially agglomerative methods).
- **Can work well for data with complex shapes** (depending on the linkage criterion).

**VIII. Disadvantages of Hierarchical Clustering**

- **Computational complexity:** Can be computationally expensive, especially for large datasets. Agglomerative clustering typically has a time complexity of O(n3) in a naive implementation, although this can be reduced to O(n2logn) with more efficient algorithms. Divisive clustering can also be computationally intensive.
- **Memory requirements:** Requires storing the proximity matrix, which can be O(n2) in size.
- **Sensitive to the choice of distance metric and linkage criterion:** Different choices can lead to significantly different results, and there's often no clear "best" choice.
- **Can be sensitive to noise and outliers:** These can affect the cluster merging/splitting decisions.
- **Difficult to handle large clusters efficiently.**

- **Once a merge or split is made, it cannot be undone.** This "greedy" nature can lead to suboptimal results if early decisions are poor.
- **May not perform as well as partitional methods (like K-Means) for large, well-separated, spherical clusters.**

**IX. Important Considerations**

- **Feature Scaling:** As with K-Means, scaling features is often important to ensure that variables with larger ranges do not dominate the distance calculations.
- **Choosing the Right Linkage:** The choice of linkage should be guided by the expected shape and structure of the clusters in your data and the goals of your analysis.
- **Validating the Clusters:** After obtaining the hierarchical clustering, it's important to evaluate the quality and interpretability of the resulting clusters using appropriate metrics or domain knowledge.

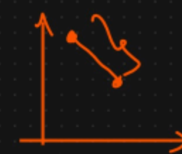
| Feature | Hierarchical Clustering | K-Means Clustering |
|---|---|---|
| **Type** | Unsupervised, hierarchical | Unsupervised, partition-based |
| **Cluster Structure** | Tree-like (dendrogram) | Flat (non-overlapping groups) |
| **Need to Specify K** | ❌ Not required (can cut dendrogram) | ✅ Must specify K beforehand |
| **Scalability** | Slower ($O(n^2)$) | Fast, scalable to large datasets |
| **Cluster Shape** | Works with arbitrary shapes | Best for spherical clusters |
| **Deterministic** | ✅ Yes (given same linkage/distance) | ❌ No (random initialization) |
| **Merge Reversal** | Not allowed | Not applicable |
| **Visualization** | Dendrogram | Scatter plot |
| **Performance with Noise** | Sensitive | Moderately sensitive |

# K Means Vs Hierarichal Clustering

## Scalability And Flexibility

① Dataset size → Huge → K Means

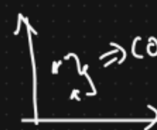
Small → Hierarichal Clustering



⇓

② KMean → Numerical data

Hierarichal clustering → Variety of data.



⇒ cosine similarity

③ Centroids → Elbow method → No. of centroids

→ No. of clusters