CS313 - Data Mining
Protein Sequence Clustering using Unsupervised Learning

Cong-Chien Hoang, Thanh-Dang Phan, Dang Tran, Dao Duong, Quang-Dat Tran, Huu-Duc Nguyen

Supervisor: Vo Nguyen Le Duy

University of Information Technology, VNU-HCM

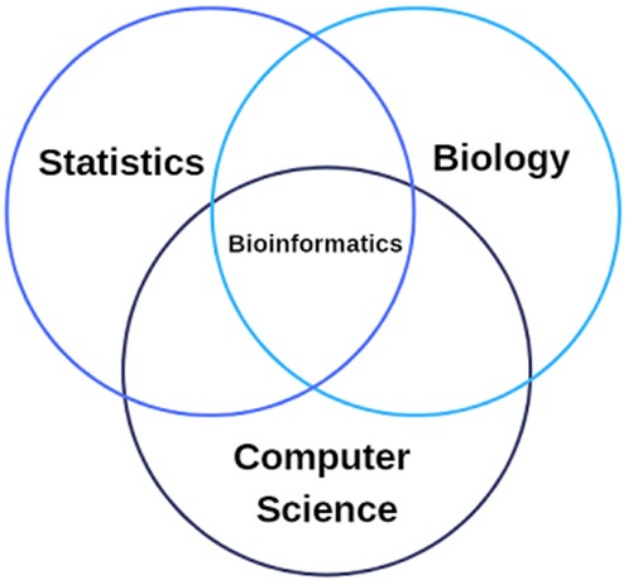
Thursday 20th March, 2025

Contents

Introduction

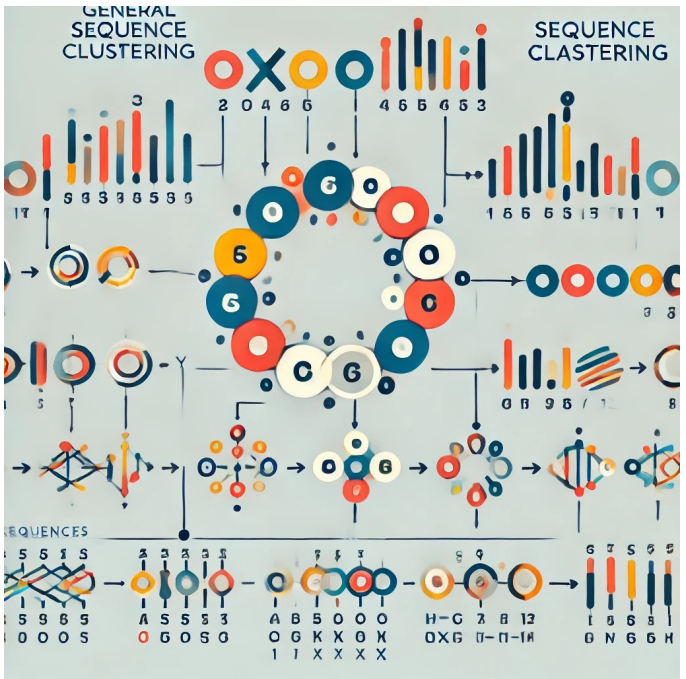
Bioinformatics

- Bioinformatics, as related to genetics and genomics, is a scientific subdiscipline that involves using computer technology to collect, store, analyze and disseminate biological data and information, such as DNA and amino acid sequences.



Sequence Clustering

- Sequence clustering is a technique used to group similar sequences of events together. The similarity between sequences is often determined based on certain metrics such as the order of events, the frequency of events, or the time between events.



Protein Sequence Clustering

Protein Sequence Clustering

- Proteins are fundamental molecules that drive most biological processes within a cell. Identifying, annotating, and characterizing proteins is crucial in bioinformatics.

Protein Sequence Clustering

- Proteins are fundamental molecules that drive most biological processes within a cell. Identifying, annotating, and characterizing proteins is crucial in bioinformatics.
- In Systems Biology, large-scale protein sequence clustering plays a vital role in detecting homology, orthology, protein families, shared domains, and functional similarities.

Protein Sequence Clustering

- Proteins are fundamental molecules that drive most biological processes within a cell. Identifying, annotating, and characterizing proteins is crucial in bioinformatics.
- In Systems Biology, large-scale protein sequence clustering plays a vital role in detecting homology, orthology, protein families, shared domains, and functional similarities.
- This project focuses on clustering protein sequences based on their similarity, enabling the identification of biologically relevant groups and relationships among proteins.

Theoretical Background

Protein

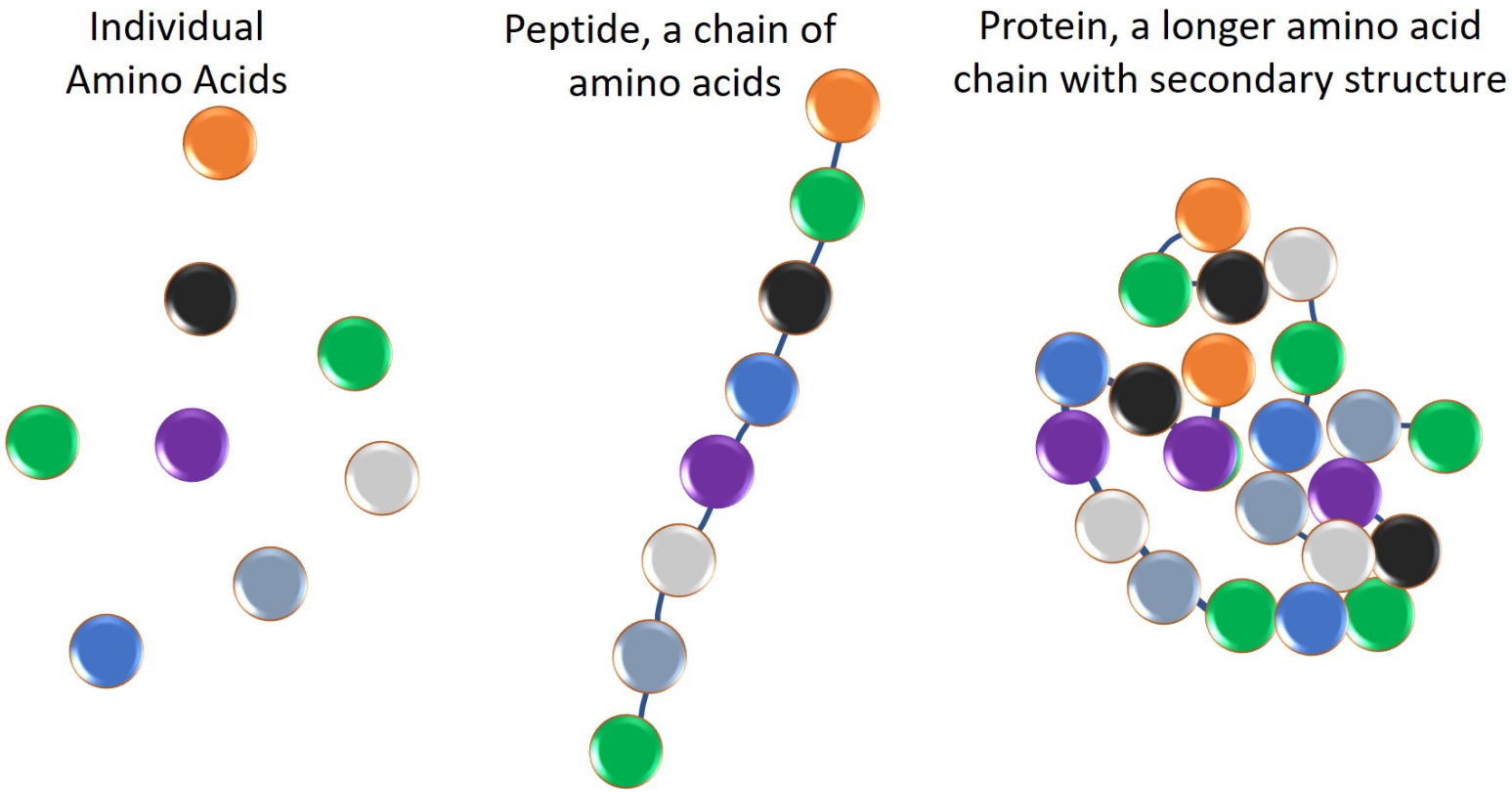
Protein

- Proteins are complex molecules that play many important roles in the body. They are critical to most of the work done by cells and are required for the structure, function and regulation of the body's tissues and organs.

Protein

- Proteins are complex molecules that play many important roles in the body. They are critical to most of the work done by cells and are required for the structure, function and regulation of the body's tissues and organs.
- A protein is made up of one or more long, folded chains of amino acids (each called a polypeptide), whose sequences are determined by the DNA sequence of the protein-encoding gene.

Protein


Individual
Amino Acids
Peptide, a chain of
amino acids
Protein, a longer amino acid
chain with secondary structure

Overview of Clustering

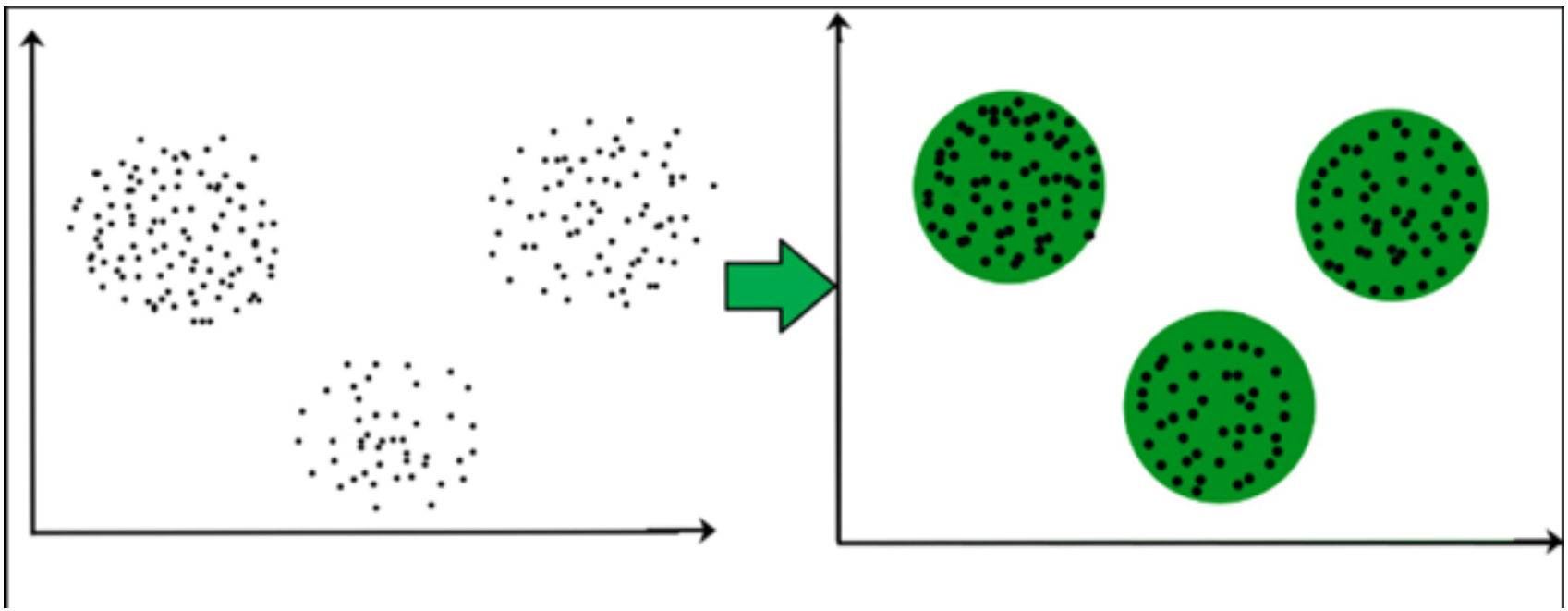
Overview of Clustering

- Clustering is an unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other. (If the examples are labeled, this kind of grouping is called classification).

Overview of Clustering

- Clustering is an unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other. (If the examples are labeled, this kind of grouping is called classification).
- Think of it as you have a dataset of customers' shopping habits. Clustering can help you group customers with similar purchasing behaviors, which can then be used for targeted marketing, product recommendations, or customer segmentation.

Overview of Clustering



Type of Clustering Algorithms

Density-based

Type of Clustering Algorithms

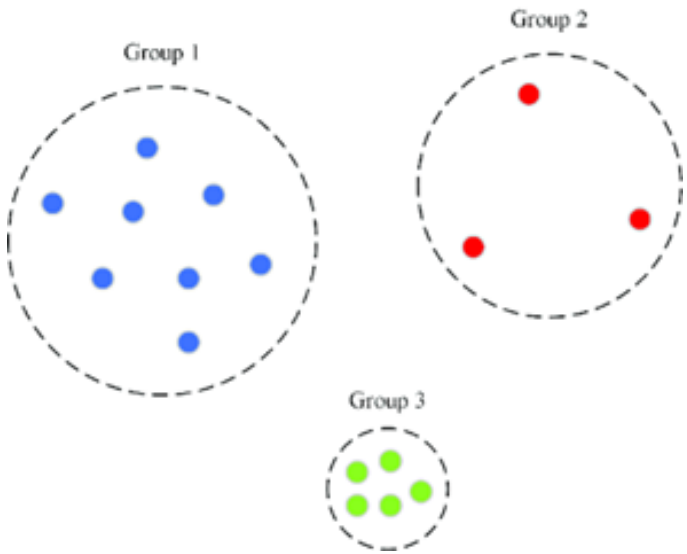
Density-based

- In density-based clustering, data is grouped by areas of high concentrations of data points surrounded by areas of low concentrations of data points. Basically the algorithm finds the places that are dense with data points and calls those clusters.
- The great thing about this is that the clusters can be any shape. You aren't constrained to expected conditions.

Type of Clustering Algorithms

Density-based

- In density-based clustering, data is grouped by areas of high concentrations of data points surrounded by areas of low concentrations of data points. Basically the algorithm finds the places that are dense with data points and calls those clusters.
- The great thing about this is that the clusters can be any shape. You aren't constrained to expected conditions.



Type of Clustering Algorithms

Distribution-based

Type of Clustering Algorithms

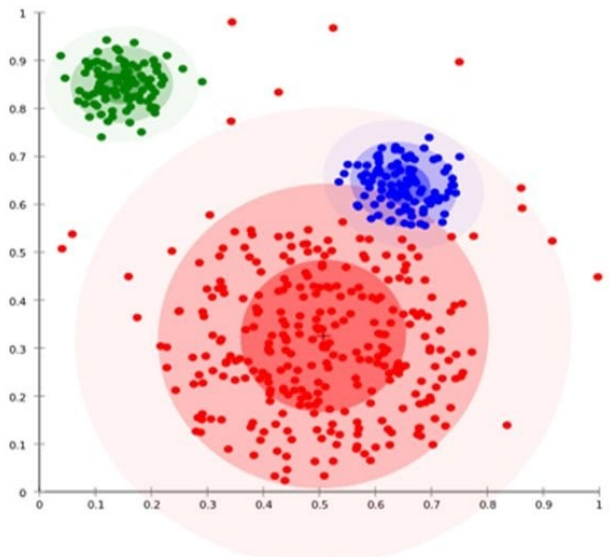
Distribution-based

- With a distribution-based clustering approach, all of the data points are considered parts of a cluster based on the probability that they belong to a given cluster.
- It works like this: there is a center-point, and as the distance of a data point from the center increases, the probability of it being a part of that cluster decreases.
- If you aren't sure of how the distribution in your data might be, you should consider a different type of algorithm.

Type of Clustering Algorithms

Distribution-based

- With a distribution-based clustering approach, all of the data points are considered parts of a cluster based on the probability that they belong to a given cluster.
- It works like this: there is a center-point, and as the distance of a data point from the center increases, the probability of it being a part of that cluster decreases.
- If you aren't sure of how the distribution in your data might be, you should consider a different type of algorithm.



Type of Clustering Algorithms

Centroid-based

Type of Clustering Algorithms

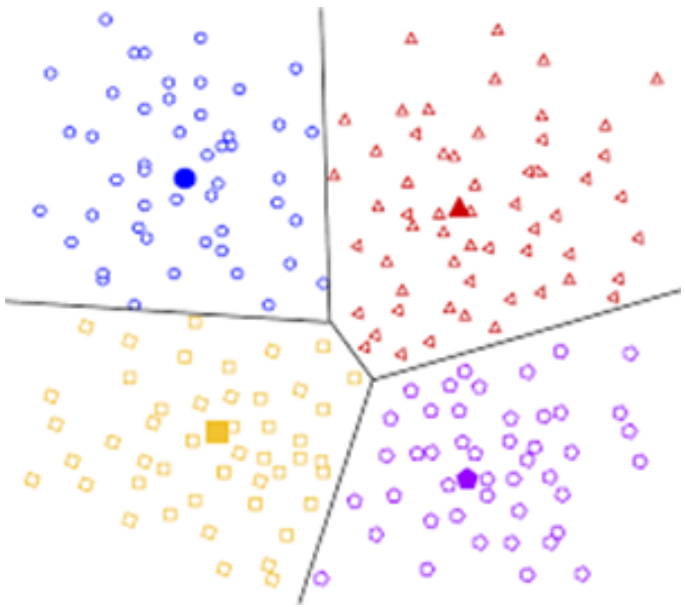
Centroid-based

- It's a little sensitive to the initial parameters, but it's fast and efficient
- These types of algorithms separate data points based on multiple centroids in the data.
- Each data point is assigned to a cluster based on its squared distance from the centroid. This is the most commonly used type of clustering.

Type of Clustering Algorithms

Centroid-based

- It's a little sensitive to the initial parameters, but it's fast and efficient
- These types of algorithms separate data points based on multiple centroids in the data.
- Each data point is assigned to a cluster based on its squared distance from the centroid. This is the most commonly used type of clustering.



Type of Clustering Algorithms

Hierarchical-based

Type of Clustering Algorithms

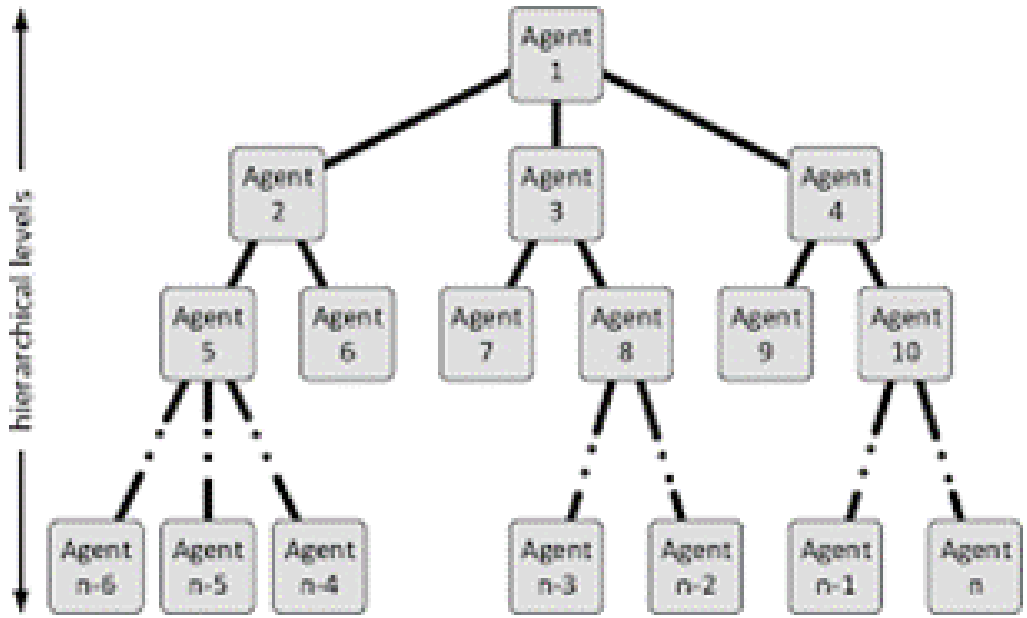
Hierarchical-based

- Hierarchical-based clustering is typically used on hierarchical data, like you would get from a company database or taxonomies. It builds a tree of clusters so everything is organized from the top-down.

Type of Clustering Algorithms

Hierarchical-based

- Hierarchical-based clustering is typically used on hierarchical data, like you would get from a company database or taxonomies. It builds a tree of clusters so everything is organized from the top-down.



Application in Bioinformatics

Application in Bioinformatics

- Sequence Analysis
- Gene Expression Analysis
- Protein Structure Analysis
- Personalized Medicine
- Metagenomics

Sequence Alignment Methods

Sequence Alignment Methods

- Sequence alignment is considered the most essential step in comparing biological sequences.

Sequence Alignment Methods

- Sequence alignment is considered the most essential step in comparing biological sequences.
- Sequence alignment arranges two or more nucleotide or amino acid sequences to identify regions of similarity between the sequences. These regions of similarity are helpful in understanding the functional, structural, and evolutionary relationships between the sequences.

Global Alignment (Needleman-Wunsch Algorithm)

Global Alignment (Needleman-Wunsch Algorithm)

- **Global alignment:** Global alignment is a method of comparing two sequences, which aligns the entire length of the sequences by maximizing the overall similarity. This method is used when comparing sequences that are of the same length.

Global Alignment (Needleman-Wunsch Algorithm)

- **Global alignment:** Global alignment is a method of comparing two sequences, which aligns the entire length of the sequences by maximizing the overall similarity. This method is used when comparing sequences that are of the same length.

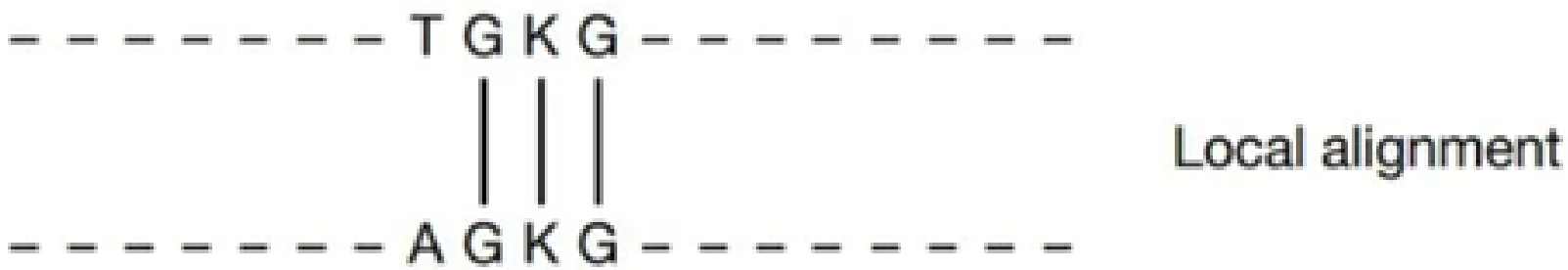
Local Alignment (Smith-Waterman Algorithm)

Local Alignment (Smith-Waterman Algorithm)

- **Local alignment:** In local alignment, instead of attempting to align the entire length of the sequences, only the regions with the highest density of matches are aligned. This is useful for identifying short conserved regions in protein or nucleotide sequences.

Local Alignment (Smith-Waterman Algorithm)

- **Local alignment:** In local alignment, instead of attempting to align the entire length of the sequences, only the regions with the highest density of matches are aligned. This is useful for identifying short conserved regions in protein or nucleotide sequences.



Clustering Algorithms

K-Means Algorithm

K-Means Algorithm

Introduction

K-Means Algorithm

Introduction

- K-Means is a clustering algorithm based on the partition-based method.

K-Means Algorithm

Introduction

- K-Means is a clustering algorithm based on the partition-based method.
- The algorithm attempts to divide data into k clusters such that the data points within the same cluster have the highest similarity and the greatest difference from other clusters.

K-Means Algorithm

Introduction

- K-Means is a clustering algorithm based on the partition-based method.
- The algorithm attempts to divide data into k clusters such that the data points within the same cluster have the highest similarity and the greatest difference from other clusters.
- The quality of the clusters is evaluated based on the distance between the data points and the cluster center (called the centroid) — typically using the Euclidean distance.

K-Means Algorithm

K-Means Algorithm

Objective

K-Means Algorithm

Objective

The goal of the algorithm is to minimize the Sum of Squared Errors (SSE):

$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_j} \| x_j - \mu_i \|_2^2$$

K-Means Algorithm

Objective

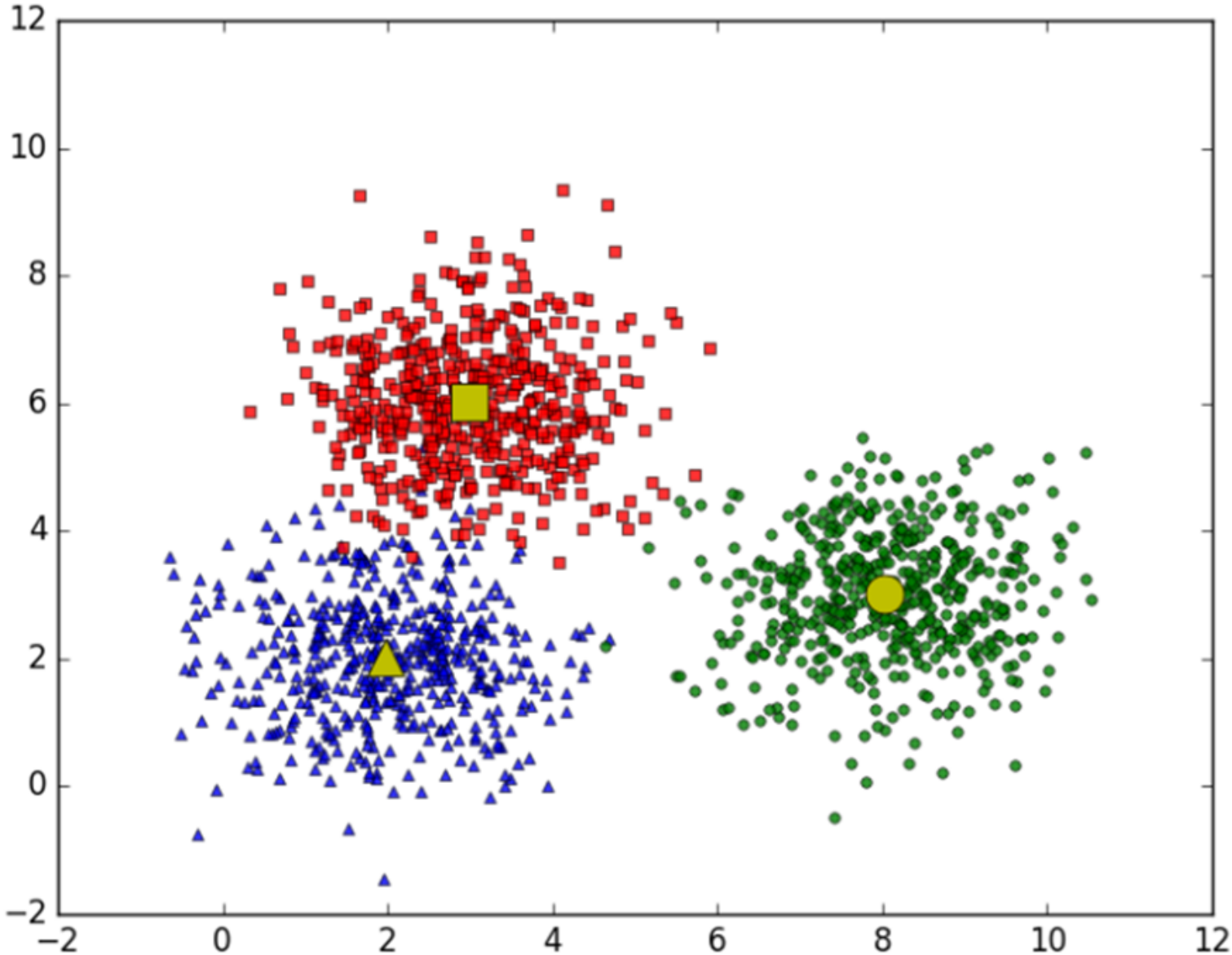
The goal of the algorithm is to minimize the Sum of Squared Errors (SSE):

$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_j} \| x_j - \mu_i \|_2^2$$

Where:

- x_j is a data point in the cluster C_i
- μ_i is the centroid of the cluster C_i
- $\| x_j - \mu_i \|_2$ is the Euclidean distance between the data point and the cluster centroid

K-Means Algorithm



K-Means Algorithm

K-Means Algorithm

Advantages

- Simple, easy to understand, and easy to implement.
- Fast computation speed $\rightarrow$ Effective for large datasets.
- Easily scalable to large problems using acceleration techniques.
- Effective when clusters have a spherical shape.

K-Means Algorithm

Advantages

- Simple, easy to understand, and easy to implement.
- Fast computation speed $\rightarrow$ Effective for large datasets.
- Easily scalable to large problems using acceleration techniques.
- Effective when clusters have a spherical shape.

Disadvantages

- Requires specifying the number of clusters k in advance.
- Sensitive to outliers.
- Results depend on the initial cluster center positions (may get stuck in local minima).
- Difficult to cluster data with irregular sizes and shapes.

Hierarchical Clustering

Hierarchical Clustering

Introduction

Hierarchical Clustering

Introduction

- Hierarchical Clustering is a clustering algorithm based on a hierarchical structure, where data is organized into clusters in the form of a tree.

Hierarchical Clustering

Introduction

- Hierarchical Clustering is a clustering algorithm based on a hierarchical structure, where data is organized into clusters in the form of a tree.
- This process can be represented by a dendrogram (a clustering tree).

Hierarchical Clustering

Introduction

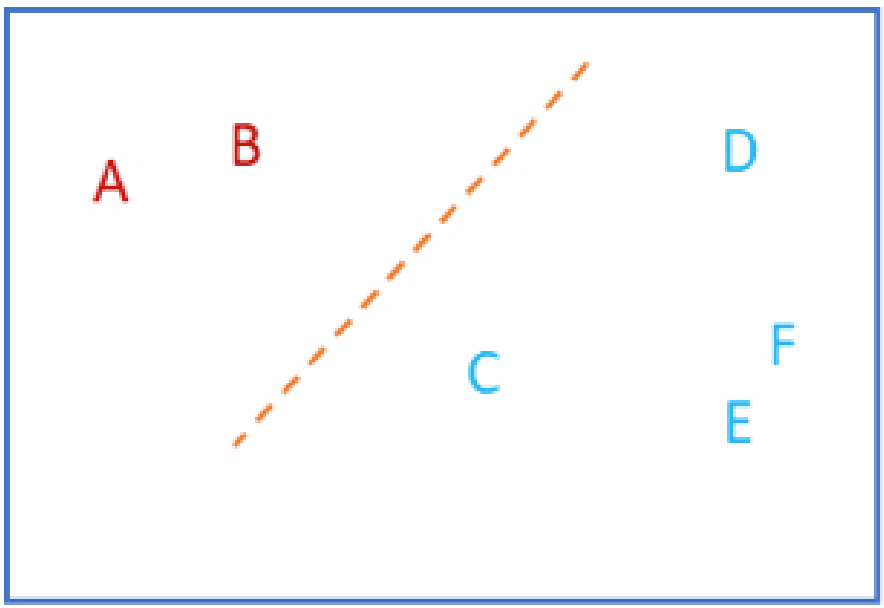
- Hierarchical Clustering is a clustering algorithm based on a hierarchical structure, where data is organized into clusters in the form of a tree.
- This process can be represented by a dendrogram (a clustering tree).
- It creates a hierarchical system, allowing users to choose an appropriate level of detail by cutting the dendrogram at a certain point.

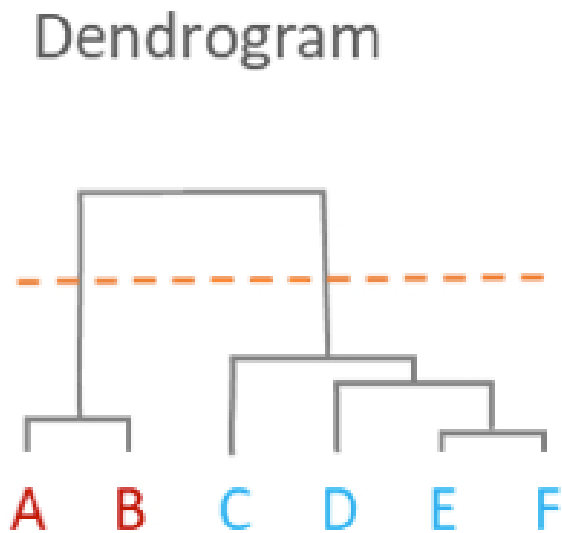
Hierarchical Clustering

Introduction

- Hierarchical Clustering is a clustering algorithm based on a hierarchical structure, where data is organized into clusters in the form of a tree.
- This process can be represented by a dendrogram (a clustering tree).
- It creates a hierarchical system, allowing users to choose an appropriate level of detail by cutting the dendrogram at a certain point.
- Hierarchical Clustering has two main methods: **Agglomerative Clustering** and **Divisive Clustering**.

Hierarchical Clustering





Hierarchical Clustering

Hierarchical Clustering

Agglomerative Clustering

- A bottom-up clustering method. Each data point starts as a separate cluster. The two closest clusters are merged, and this process continues until all data points belong to a single cluster.
- This process is represented by a dendrogram, which helps users select the optimal number of clusters.

Hierarchical Clustering

Agglomerative Clustering

- A bottom-up clustering method. Each data point starts as a separate cluster. The two closest clusters are merged, and this process continues until all data points belong to a single cluster.
- This process is represented by a dendrogram, which helps users select the optimal number of clusters.

Divisive Clustering

- A top-down clustering method. The data starts in a single cluster and is then recursively split into smaller clusters until each data point becomes its own cluster.

Hierarchical Clustering

Agglomerative Clustering

Hierarchical Clustering

Agglomerative Clustering

Work Flow

1. **Start with individual points:** Each data point is its own cluster.
2. **Calculate distances between clusters:** Calculate the distance between every pair of clusters.
3. **Merge the closest clusters:** Identify the two clusters with the smallest distance and merge them into a single cluster.
4. **Update distance matrix:** After merging you now have one less cluster. Recalculate the distances between the new cluster and the remaining clusters.
5. **Repeat steps 3 and 4:** Keep merging the closest clusters and updating the distance matrix until you have only one cluster left.

Hierarchical Clustering

Linkage Methods

Hierarchical Clustering

Linkage Methods

- Assuming at a specific level in the dendrogram, we have two non-overlapping intermediate clusters:

$$\mathcal{S}_1 = \{x_i^{(1)}\}_{i=1}^{N_1} \text{ and } \mathcal{S}_2 = \{x_j^{(2)}\}_{j=1}^{N_2}.$$

Hierarchical Clustering

Linkage Methods

- Assuming at a specific level in the dendrogram, we have two non-overlapping intermediate clusters:

$$\mathcal{S}_1 = \{x_i^{(1)}\}_{i=1}^{N_1} \text{ and } \mathcal{S}_2 = \{x_j^{(2)}\}_{j=1}^{N_2}.$$

- The distance between the two clusters represents the difference between them. There are several methods to determine the distance between two clusters, including: *Single Linkage*, *Complete Linkage*, *Average Linkage*, *Centroid Linkage*.

Hierarchical Clustering

Hierarchical Clustering

- **Single Linkage** calculates the distance between two clusters as the shortest distance between two points belonging to those two clusters. This method can produce elongated, chain-like clusters, leading to some clusters having non-uniform shapes.

$$d(\mathcal{S}_1, \mathcal{S}_2) = \min_{x_i \in \mathcal{S}_1, x_j \in \mathcal{S}_2} d(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)})$$

Hierarchical Clustering

- **Single Linkage** calculates the distance between two clusters as the shortest distance between two points belonging to those two clusters. This method can produce elongated, chain-like clusters, leading to some clusters having non-uniform shapes.

$$d(\mathcal{S}_1, \mathcal{S}_2) = \min_{x_i \in \mathcal{S}_1, x_j \in \mathcal{S}_2} d(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)})$$

- **Complete Linkage** measures the distance between two clusters as the farthest distance between two points from the two clusters. This helps create more compact clusters, avoiding the issue of excessively elongated clusters as seen in Single Linkage.

$$d(\mathcal{S}_1, \mathcal{S}_2) = \max_{x_i \in \mathcal{S}_1, x_j \in \mathcal{S}_2} d(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)})$$

Hierarchical Clustering

Hierarchical Clustering

- **Average Linkage**: This method calculates the average of all distances between pairs of points taken from the two clusters. We will have a total of pairs of points. Thus, the distance will be calculated as:

$$d(S_1, S_2) = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} d(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)})$$

Hierarchical Clustering

- **Average Linkage**: This method calculates the average of all distances between pairs of points taken from the two clusters. We will have a total of pairs of points. Thus, the distance will be calculated as:

$$d(S_1, S_2) = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} d(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)})$$

- **Centroid Linkage** measures the distance between two clusters as the distance between the centroids of those two clusters. This method works well when the data is uniformly distributed and can reduce sensitivity to noise.

$$d(\mathcal{S}_1, \mathcal{S}_2) = \| \mu_{\mathcal{S}_1} - \mu_{\mathcal{S}_2} \|$$

Hierarchical Clustering

Hierarchical Clustering

Advantages

- No need to specify the number of clusters.
- Produces a dendrogram for visual interpretation.
- Captures clusters of varying shapes and sizes.
- Flexible with distance metrics and linkage methods.

Hierarchical Clustering

Advantages

- No need to specify the number of clusters.
- Produces a dendrogram for visual interpretation.
- Captures clusters of varying shapes and sizes.
- Flexible with distance metrics and linkage methods.

Disadvantages

- Computationally expensive for large datasets.
- Sensitive to noise and outliers.
- Irreversible merges can lead to sub-optimal results.
- Results depend heavily on the choice of linkage method.

Comparison

Comparison

K-means	Hierarchical
Partitioning-based clustering	Builds a hierarchy of clusters
Need to determine the number of clusters	No need to specify the number of clusters
Computes faster with large datasets	Computationally expensive for large datasets
More sensitive to outliers	Less sensitive to outliers
Assumes clusters are spherical and of similar size	Can handle clusters of arbitrary shapes

Demo

Thank You!

Questions?