



# Introduction to Data Mining

## Chapter 7 Cluster Analysis

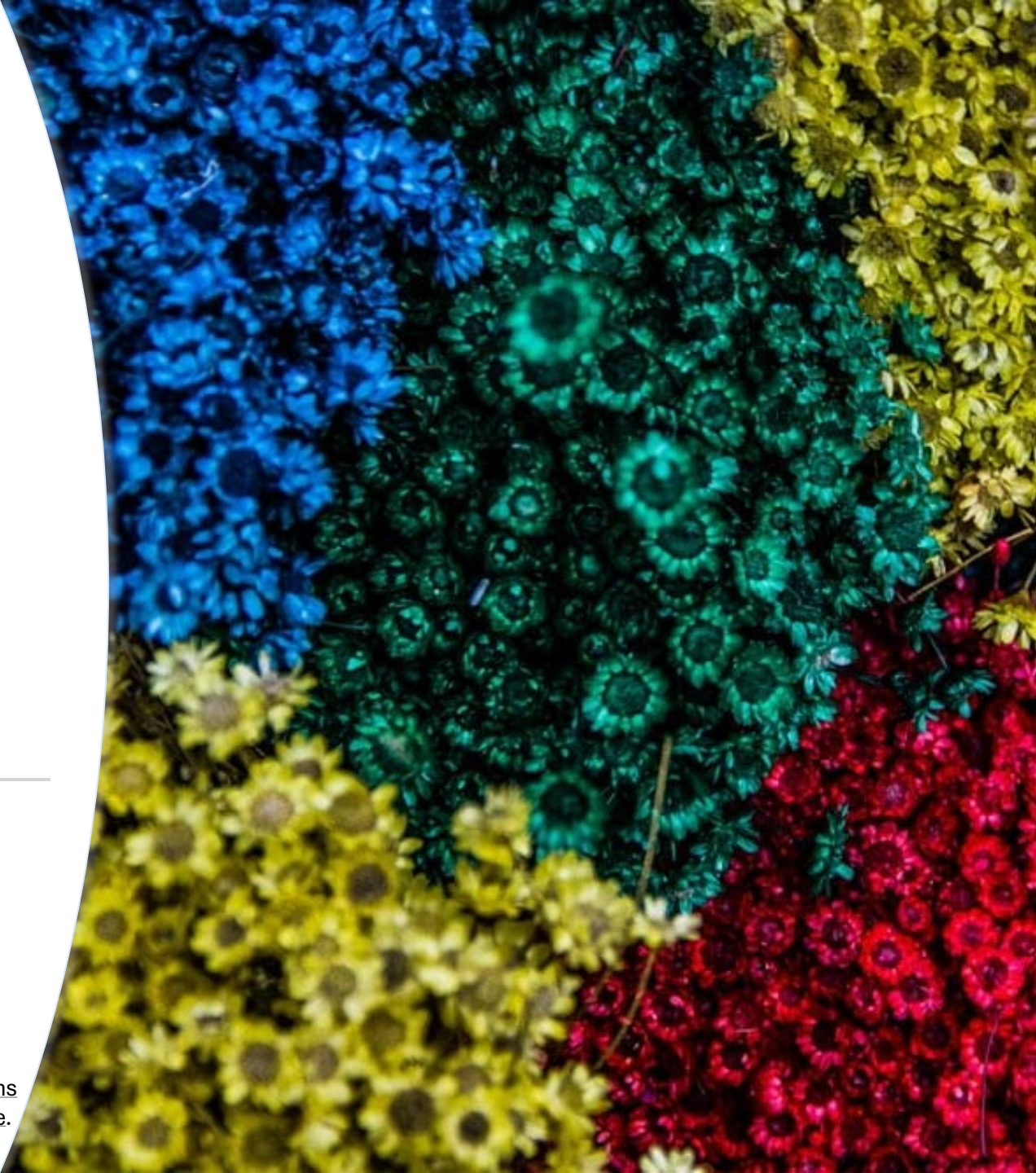
---

by Michael Hahsler

Based in Slides by Tan,  
Steinbach, Karpatne, Kumar



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).



# R Code Examples

- Available R Code examples are indicated on slides by the R logo



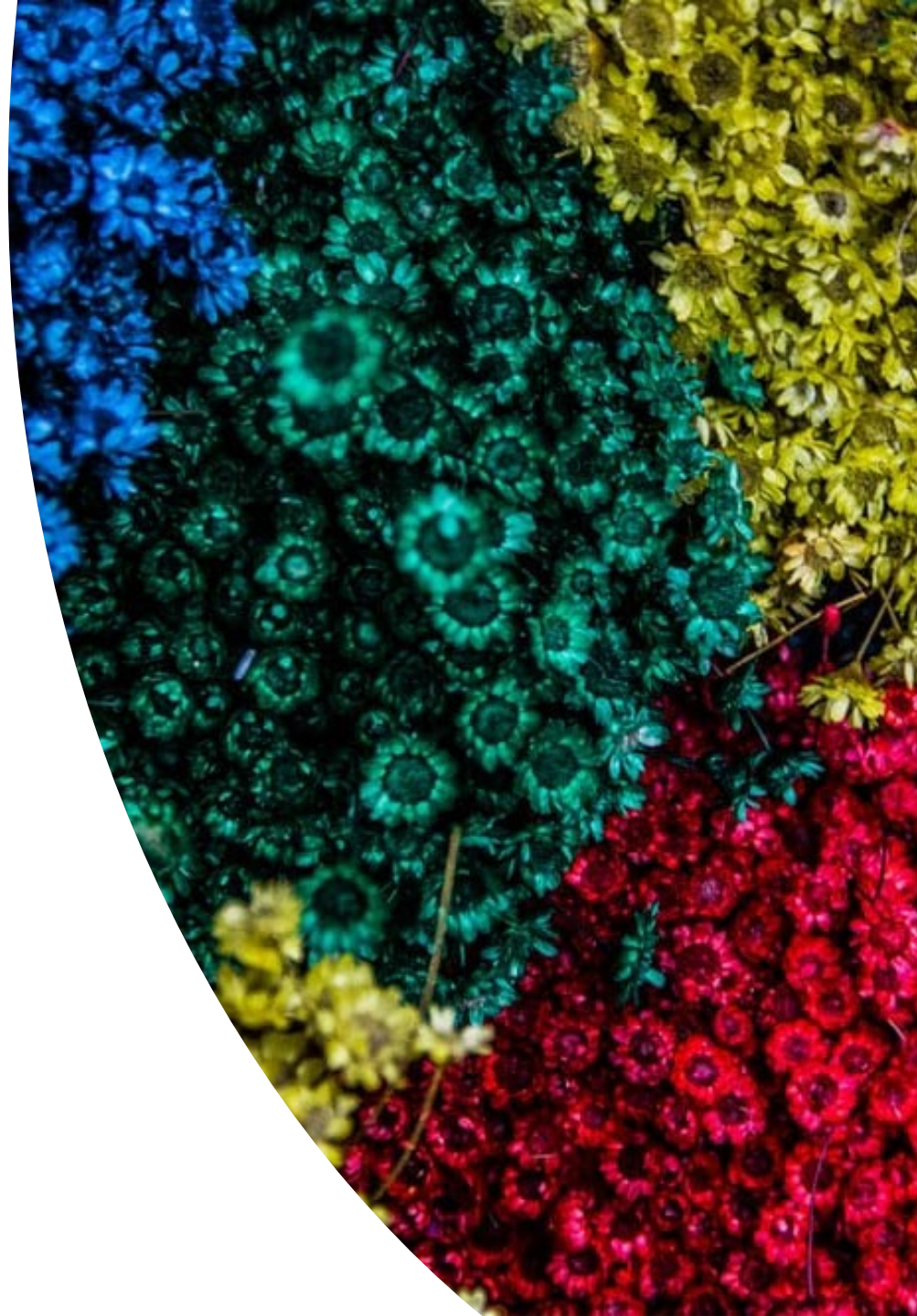
- The Examples are available at [https://mhahsler.github.io/Introduction\\_to\\_Data\\_Mining\\_R\\_Examples/](https://mhahsler.github.io/Introduction_to_Data_Mining_R_Examples/)



# Topics

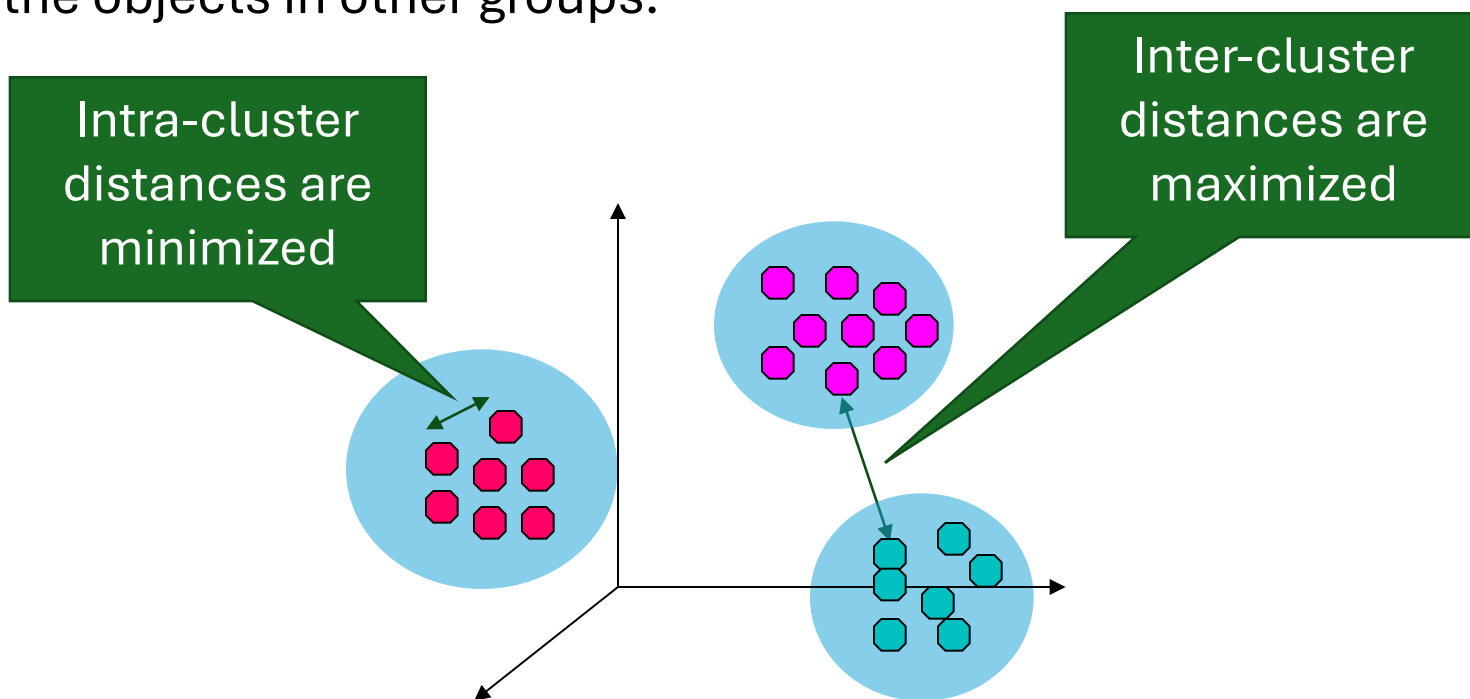
---

- **Introduction**
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation
- Outliers and Scaling Issues



# What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.



- A clustering is a set of clusters. Each cluster contains a set of points.

# Applications of Cluster Analysis

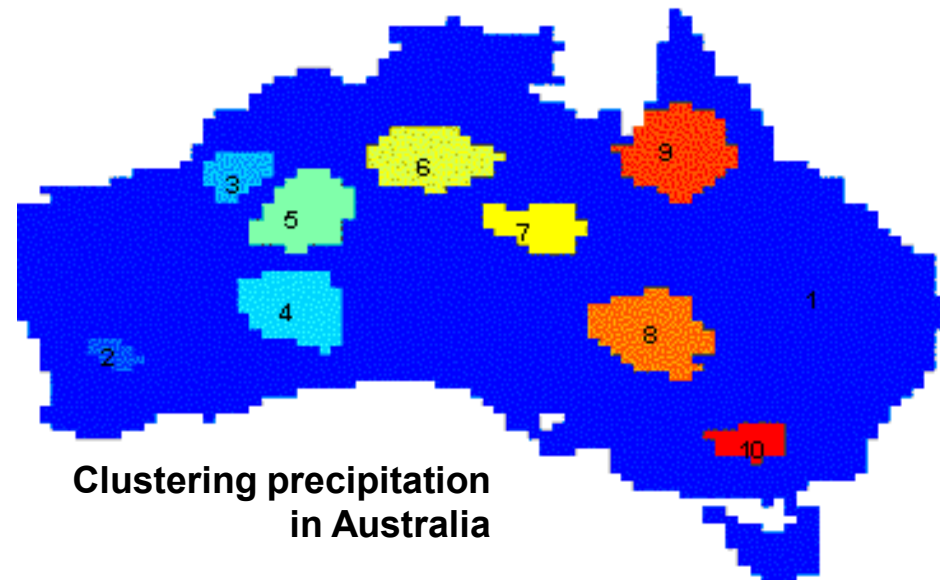
- Understanding

- Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

- Summarization

- Reduce the size of large data sets to a small number of groups.

|   | <i>Discovered Clusters</i>  | <i>Industry Group</i> |
|---|---|-----------------------|
| 1 | Applied-Matl-DOWN,Bay-Network-DOWN,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-DOWN,Tellabs-Inc-DOWN, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN | Technology1-DOWN      |
| 2 | Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN                    | Technology2-DOWN      |
| 3 | Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN  | Financial-DOWN        |
| 4 | Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP  | Oil-UP                |



# Measuring Similarity/Distances

- How do we measure **similarity/dissimilarity/distance/proximity?**
- Examples
  - Minkovsky distance: Manhattan distance, Euclidean Distance, etc.
  - Jaccard index for binary data.
  - Cosine similarity for word counts.
  - Gower's distance for mixed data (ratio/interval and nominal).
  - Correlation coefficient as similarity between variables.
- See Chapter 2 on Data.

# What is not Cluster Analysis?

→ Clustering organizes observations by descriptive features

- Supervised classification

- Uses correct class label information to learn how to predict the class label.

- Simple segmentation

- E.g., Dividing students into different registration groups alphabetically, by last name.

- Results of a query

- Groupings are a result of an external query specification.

# Clustering as Unsupervised Learning

- Examples

- Input data:  $E = x_1, x_2, \dots, x_i, \dots, x_N$ .
- We assume that the examples are produced iid (with noise and errors) from a set of  $k$  clusters  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ .
- The correct assignment is not part of the input data!

- Learning problem

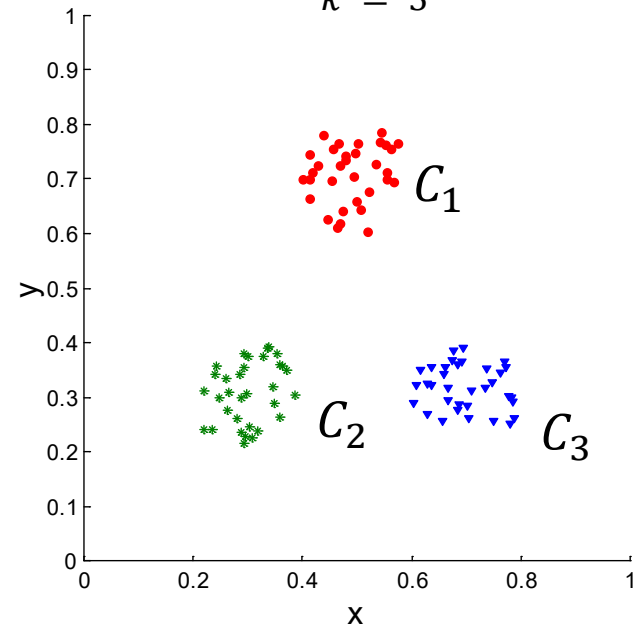
- Find an assignment function

$$y = f(x)$$

where  $y \in \mathcal{C}$  is a cluster label such that an objective function measuring the quality of the clustering is minimized.

## Example

$k = 3$



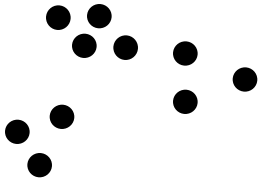


# Notion of a Cluster can be Ambiguous

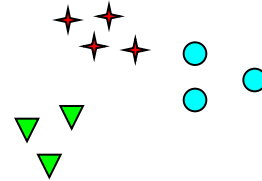
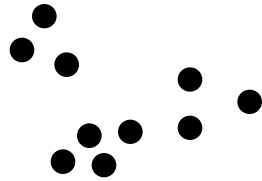


How many clusters?

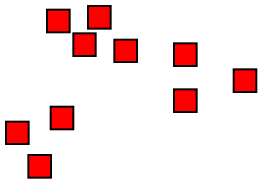
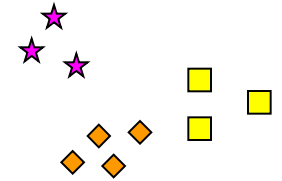
# Notion of a Cluster can be Ambiguous



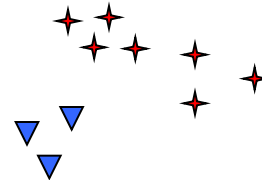
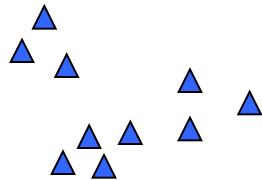
How many clusters?



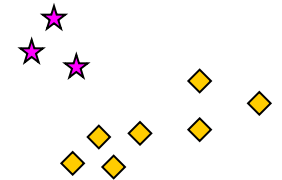
Six Clusters



Two Clusters

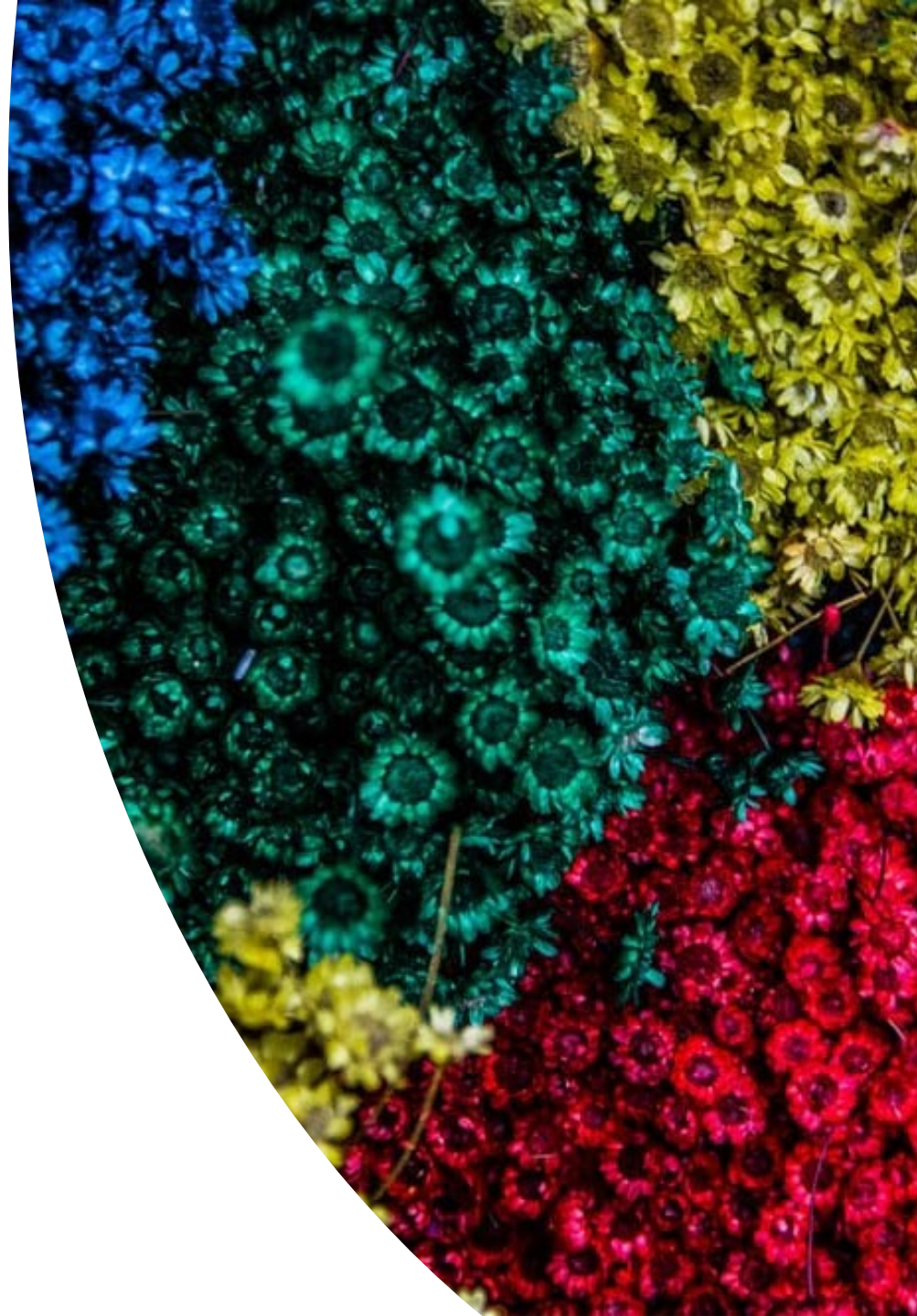


Four Clusters



# Topics

- Introduction
  - **Types of Clustering**
  - Types of Clusters
  - Clustering Algorithms
    - K-Means Clustering
    - Hierarchical Clustering
    - Density-based Clustering
  - Cluster Evaluation
    - Unsupervised Evaluation
    - Supervised Evaluation
- Outliers and Scaling Issues



# Types of Clusterings



## Partitional Clustering

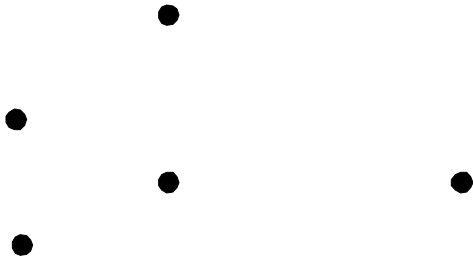
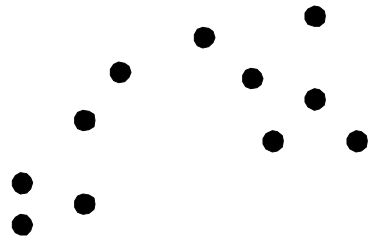
A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset



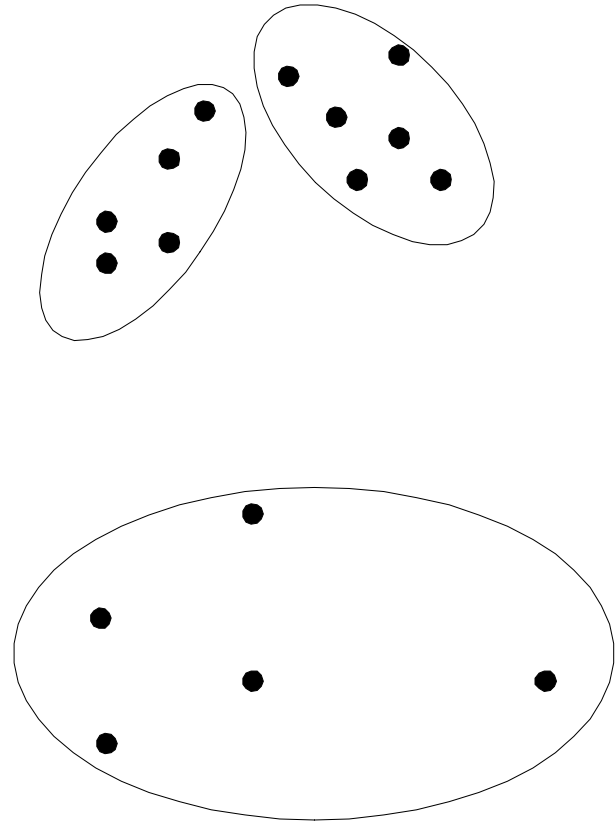
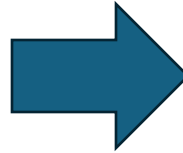
## Hierarchical clustering

A set of nested clusters organized as a hierarchical tree

# Partitional Clustering

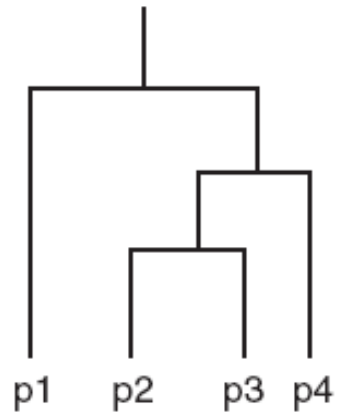


**Original Points**

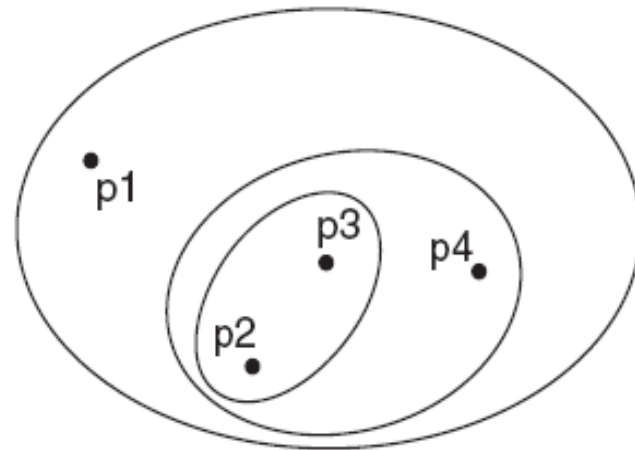


**A Partitional Clustering**

# Hierarchical Clustering



(a) Dendrogram.



(b) Nested cluster diagram.

**Figure 8.13.** A hierarchical clustering of four points shown as a dendrogram and as nested clusters.

# Other Distinctions Between Sets of Clusters



## Exclusive versus non-exclusive

In non-exclusive clusterings, points may belong to multiple clusters.



## Fuzzy versus non-fuzzy

In fuzzy clustering, a point belongs to every cluster with some membership weight between 0 and 1.

Membership weights must sum to 1.



## Partial versus complete

In some cases, we only want to cluster some of the data.

E.g. don't cluster outliers or noise data points.



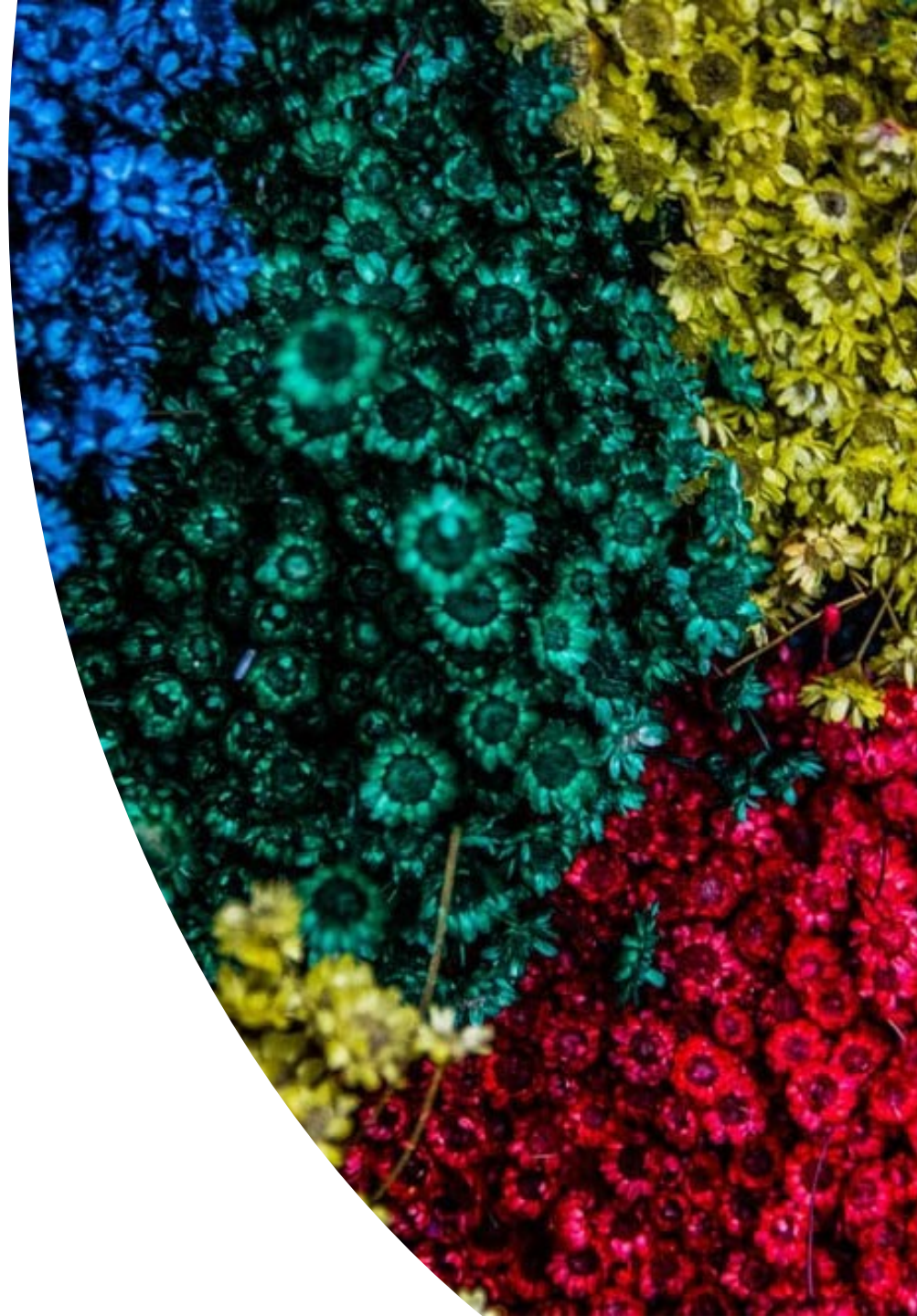
## Heterogeneous versus homogeneous

Cluster of widely different sizes, shapes, and densities

Typical clustering is: exclusive, crisp (non-fuzzy), complete, and homogeneous.

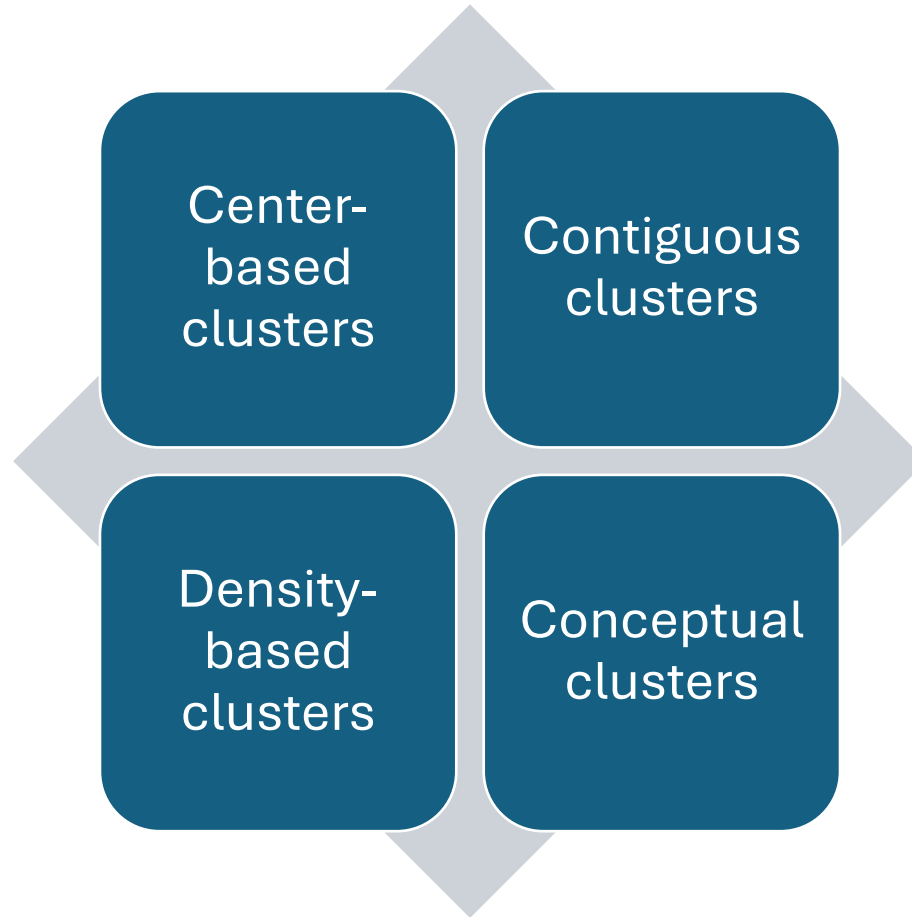
# Topics

- Introduction
  - Types of Clustering
  - **Types of Clusters**
  - Clustering Algorithms
    - K-Means Clustering
    - Hierarchical Clustering
    - Density-based Clustering
  - Cluster Evaluation
    - Unsupervised Evaluation
    - Supervised Evaluation
- Outliers and Scaling Issues

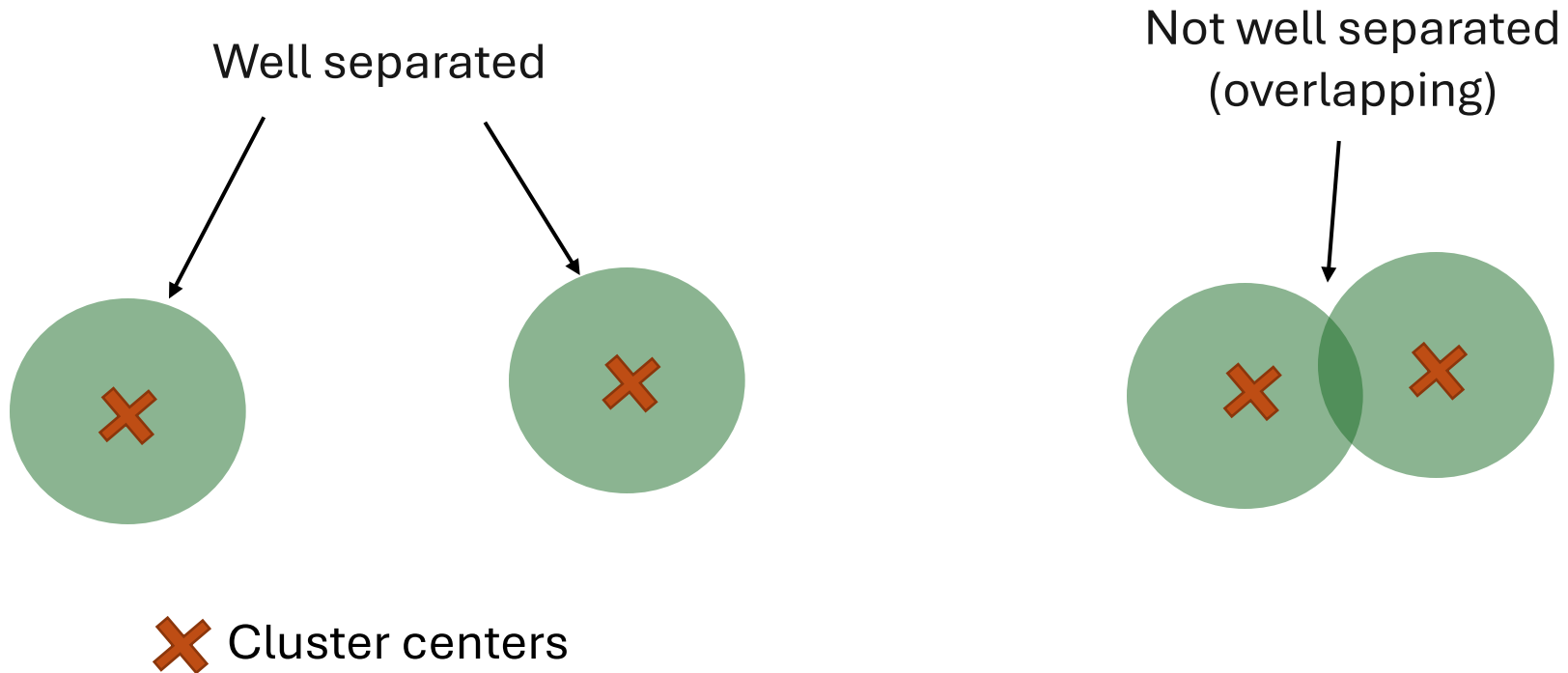




# Types of Clusters



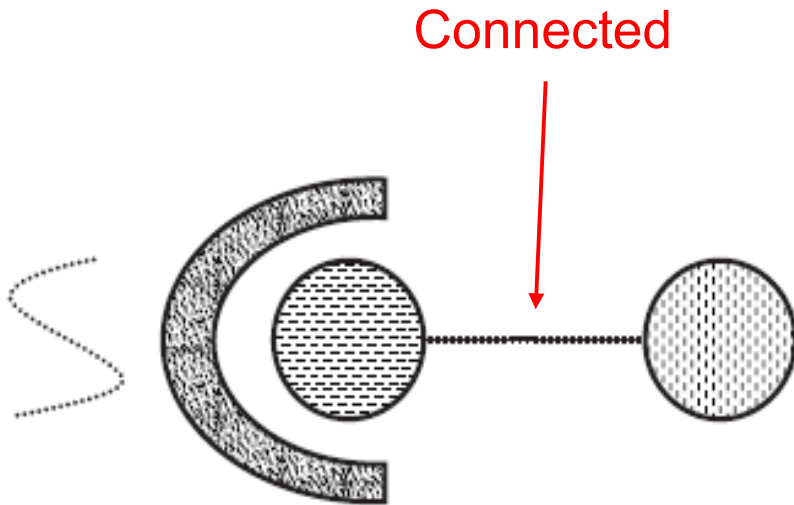
# Center-based Clusters



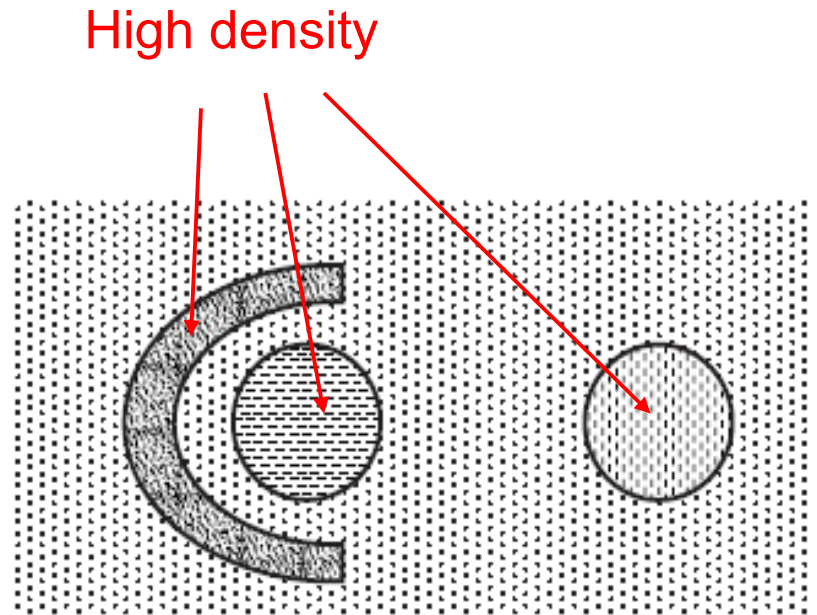
A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster

The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most “representative” point of a cluster

# Contiguous and Density-based Clusters

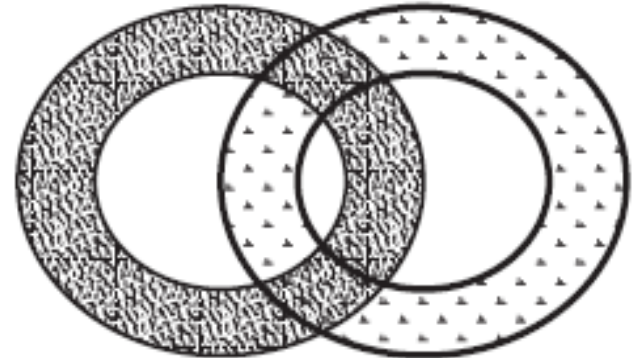
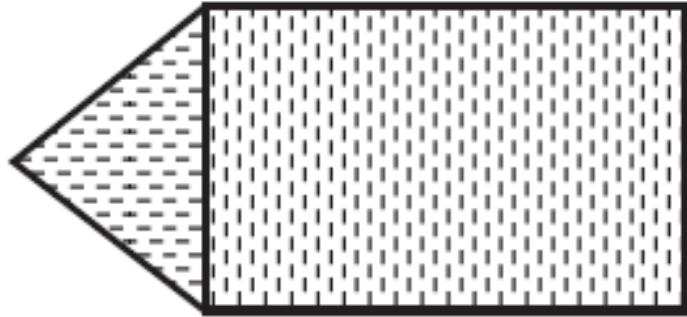


(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

# Conceptual Clusters



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

Conceptual clusters are hard to detect since they are often not:

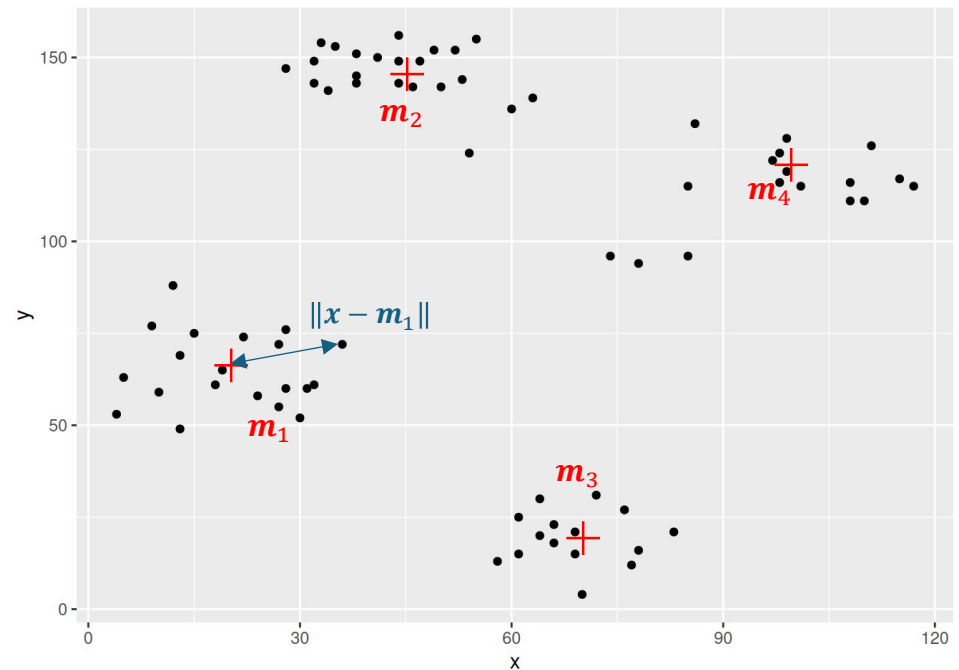
- Center-based
- Contiguity-based
- Density-based

# Clustering is an Optimization Problem

- The best clustering minimizes or maximizes an objective function.
- **Example:** Minimize the “Sum of Squared Errors” for centroid-based algorithms.

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} \|x - m_i\|^2$$

- $x$  is a data point in cluster  $C_i$ ,
- $m_i$  is the center for cluster  $C_i$  as the mean of all points in the cluster.
- $\|\cdot\|$  is the L2 norm (Euclidean distance)



# Objective Functions

## Global objective function

- Typically used in **partitional clustering**. k-means uses SSE.
- **Mixture Models** assume that the data is a 'mixture' of a number of parametric statistical distributions (e.g., a mixture of Gaussians). Maximize log-likelihood of the model.

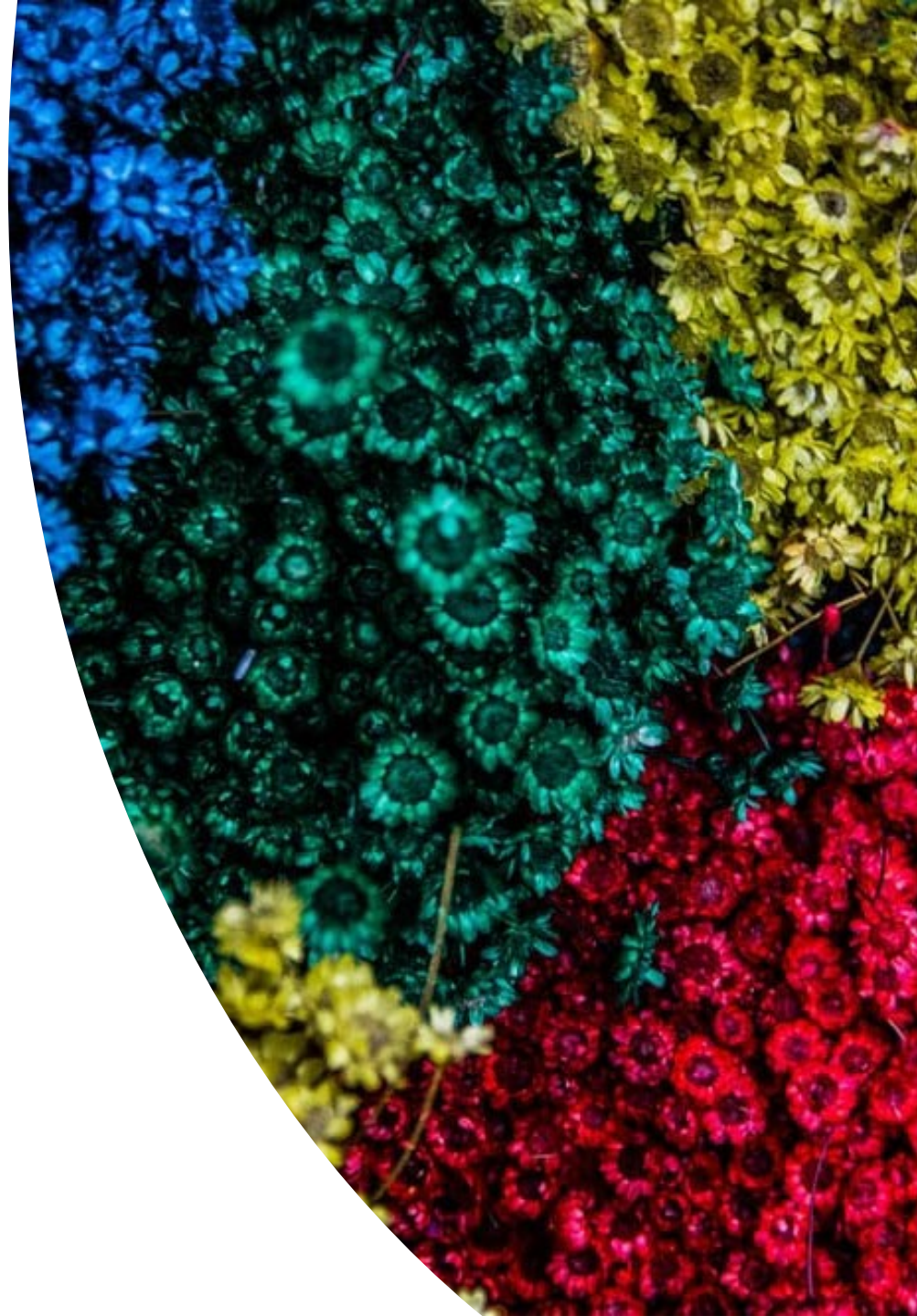
## Local objective function

- **Hierarchical clustering** algorithms typically have local objectives.
- **Density-based clustering** is based on local density estimates.
- **Graph based approaches**. Graph partitioning (e.g., min-cut) and shared nearest neighbors.

We will talk about the objective functions more when we talk about individual clustering algorithms.

# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- **Clustering Algorithms**
  - **K-Means Clustering**
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation
- Outliers and Scaling Issues



# K-means Clustering

- **Partitional clustering** approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters,  $K$ , must be specified

Lloyd's algorithm (Voronoi iteration):

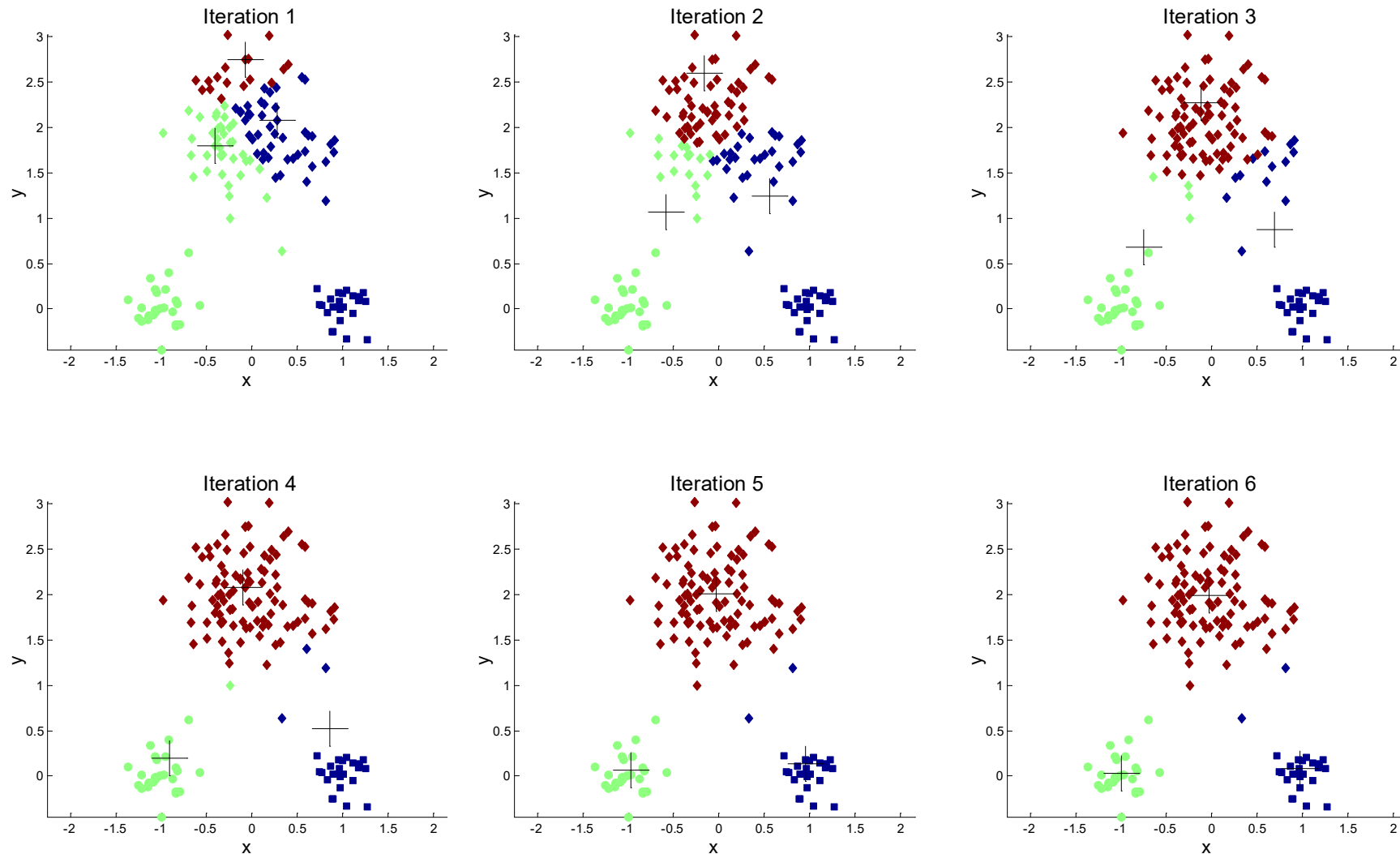
- 1: Select  $K$  points as the initial centroids.
- 2: **repeat**
- 3:     Form  $K$  clusters by assigning all points to the closest centroid.
- 4:     Recompute the centroid of each cluster.
- 5: **until** The centroids don't change



# K-means Clustering – Details

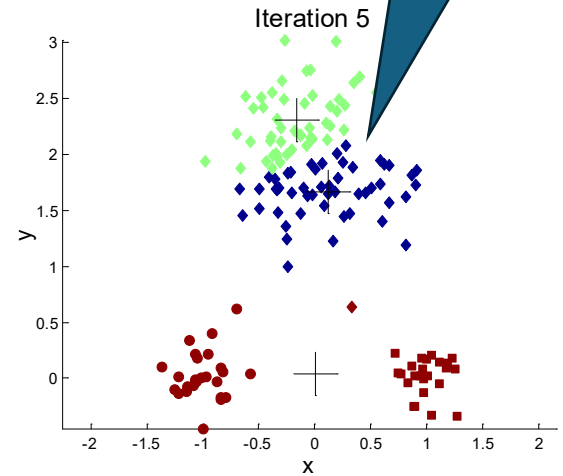
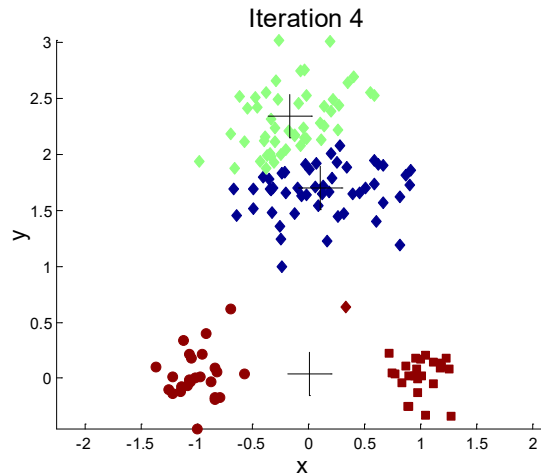
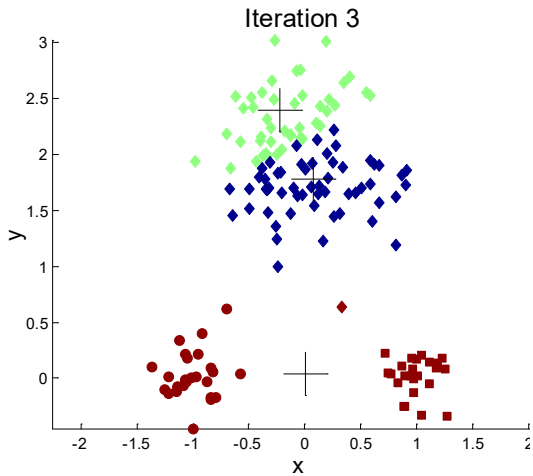
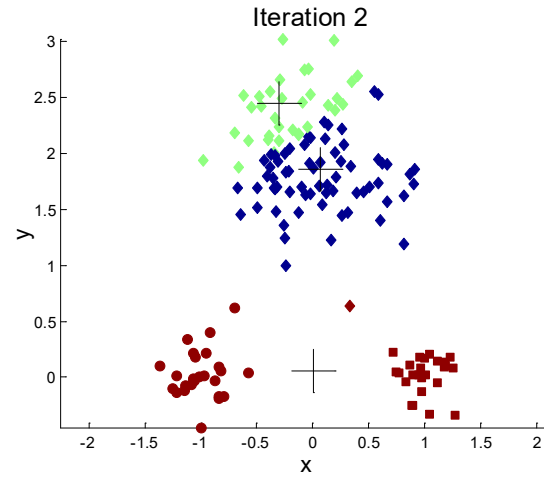
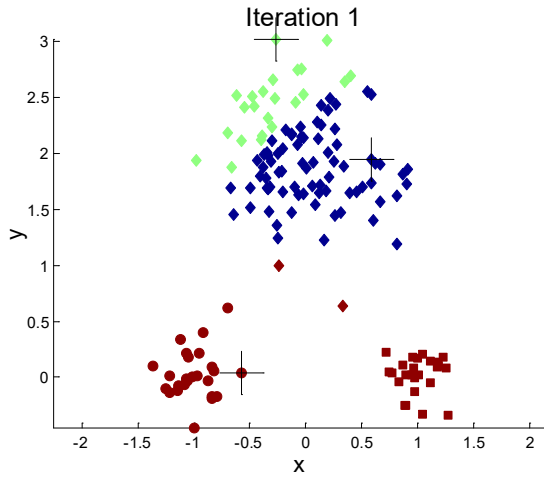
- **Initial centroids** are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is the mean of the points in the cluster.
- ‘Closeness’ is measured by **Euclidean distance**
- K-means will converge (points stop changing assignment) typically in the first few iterations (<10).
  - Sometimes the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is  $O(nKId)$ 
  - $n$  = number of points,  $K$  = number of clusters,
  - $I$  = number of iterations,  $d$  = number of attributes

# K-Means Example



→ See visualization

# Importance of Choosing Initial Centroids ...



Gets stuck in a local optimum!

# Solutions to the Initial Centroids Problem

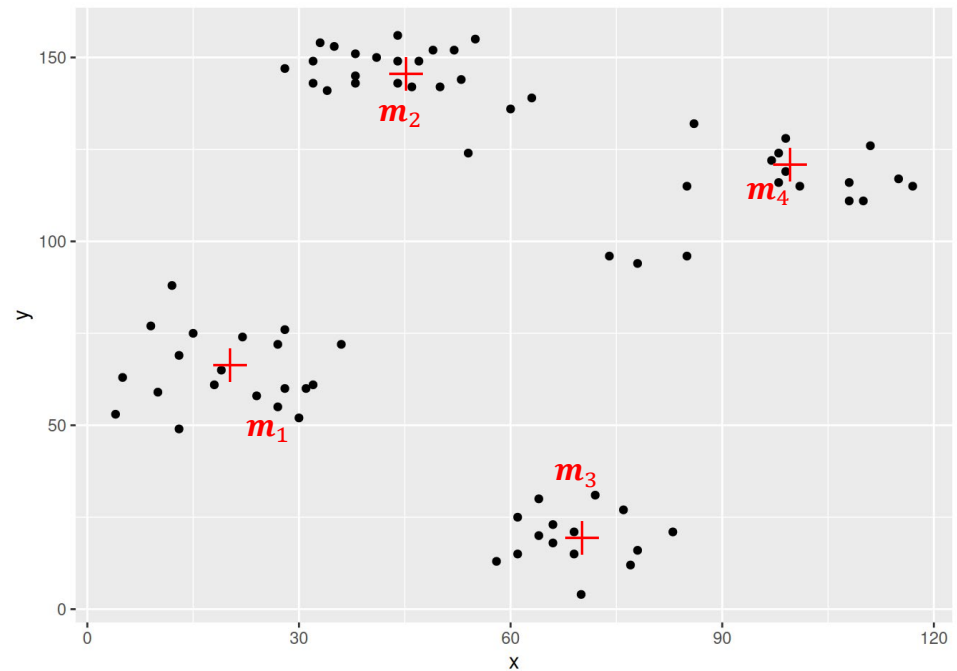
- Multiple runs. This is standard in most tools and typically helps.
- Sample and use hierarchical clustering to determine initial centroids.
- Select more than  $k$  initial centroids and then select among these initial centroids the ones that are far away from each other.

# Evaluating k-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster center
  - k-means uses a simple heuristic to minimizing the SSE.

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} \|x - m_i\|^2$$

- Given two clusterings, we can choose the one with the smallest *SSE*.
- Only compare clusterings with the same number of clusters *k*! *SSE* automatically decreases with *k*.
- *SSE* is also called the *WCSS* or *WSS*.



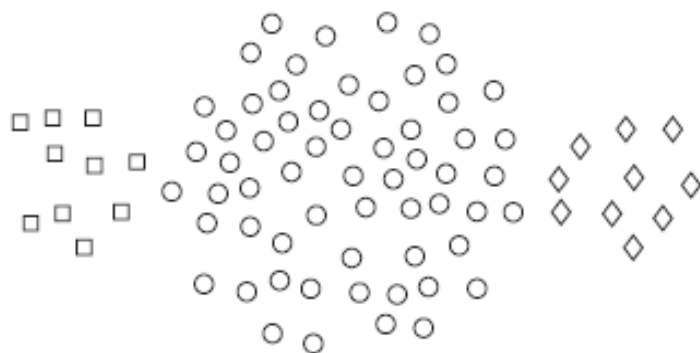
# Pre-processing and Post-processing

- Pre-processing
  - **Normalize** the data (e.g., scale to unit standard deviation)
  - **Eliminate outliers**
- Post-processing
  - **Eliminate** small clusters that may represent outliers
  - **Split** 'loose' clusters, i.e., clusters with relatively high SSE
  - **Merge** clusters that are 'close' and that have relatively low SSE

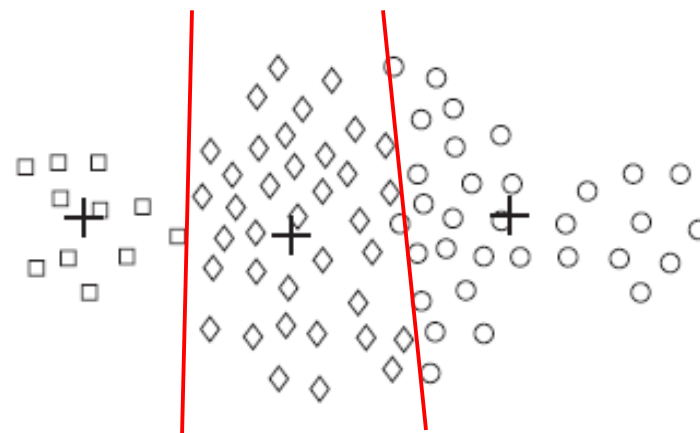
# Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.

# Limitations of K-means: Differing Sizes



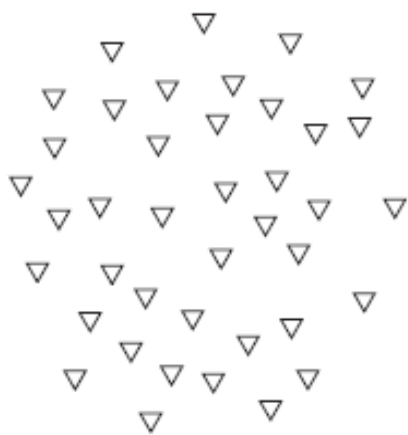
(a) Original points.



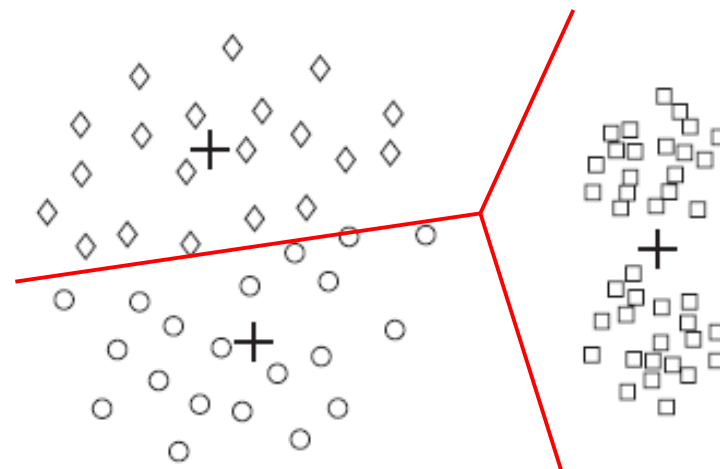
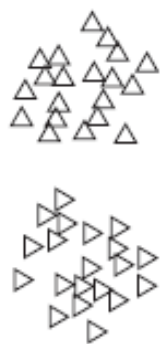
(b) Three K-means clusters.



# Limitations of K-means: Differing Density

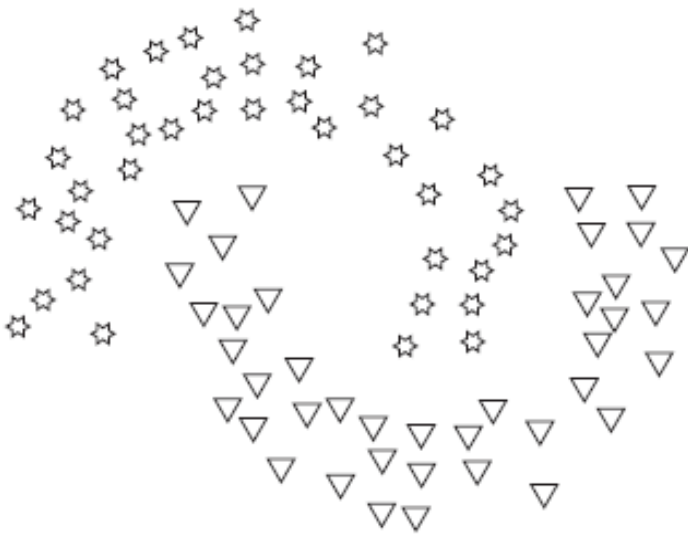


(a) Original points.

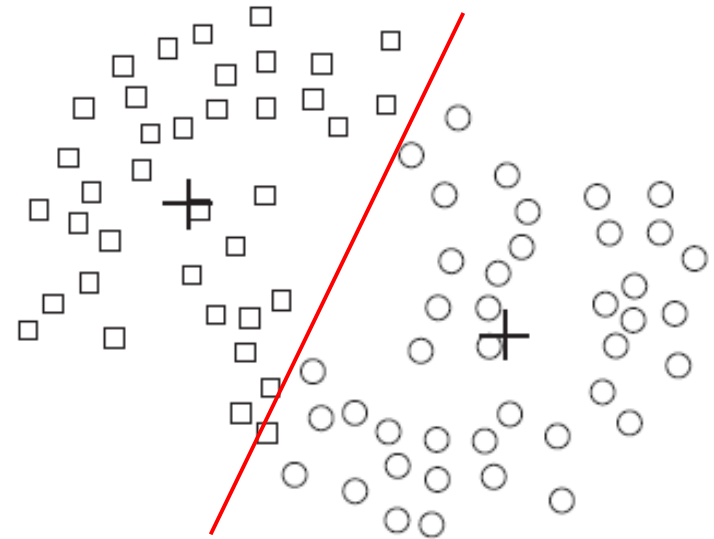


(b) Three K-means clusters.

# Limitations of K-means: Non-globular Shapes

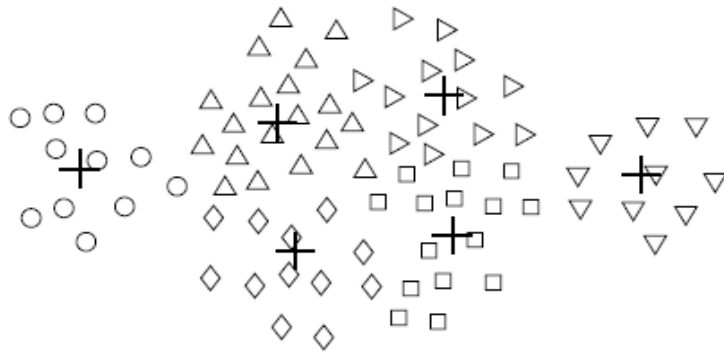


(a) Original points.

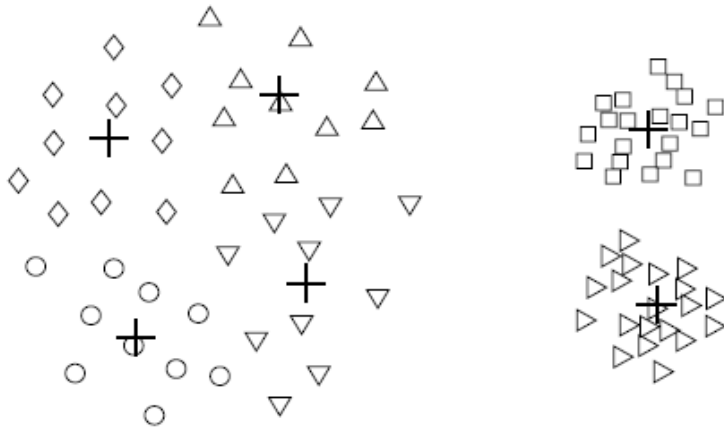


(b) Two K-means clusters.

# Overcoming K-means Limitations



(a) Unequal sizes.

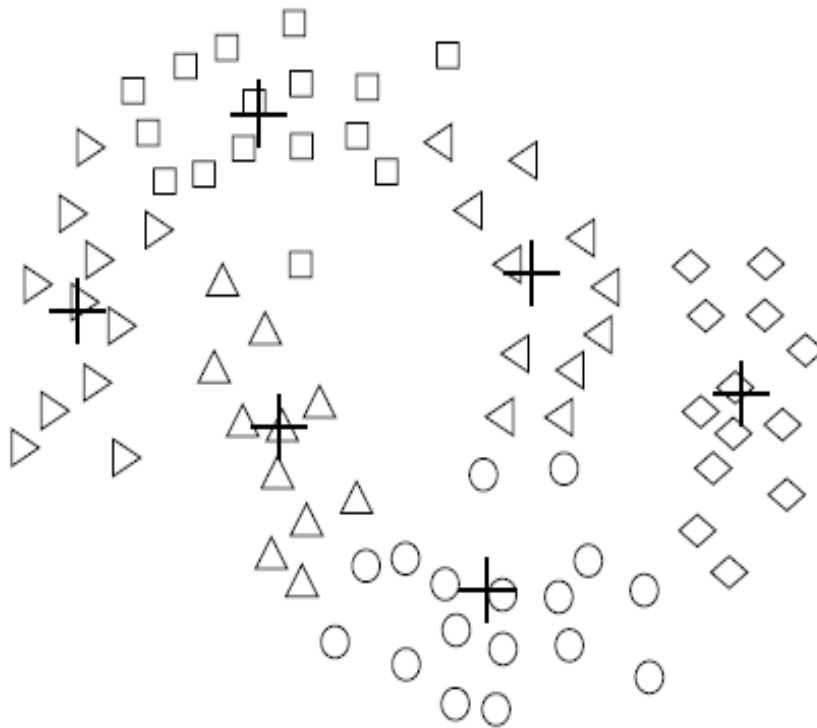


(b) Unequal densities.

Use a larger number of clusters

Several clusters represent a true Cluster and need to be merged.

# Overcoming K-means Limitations



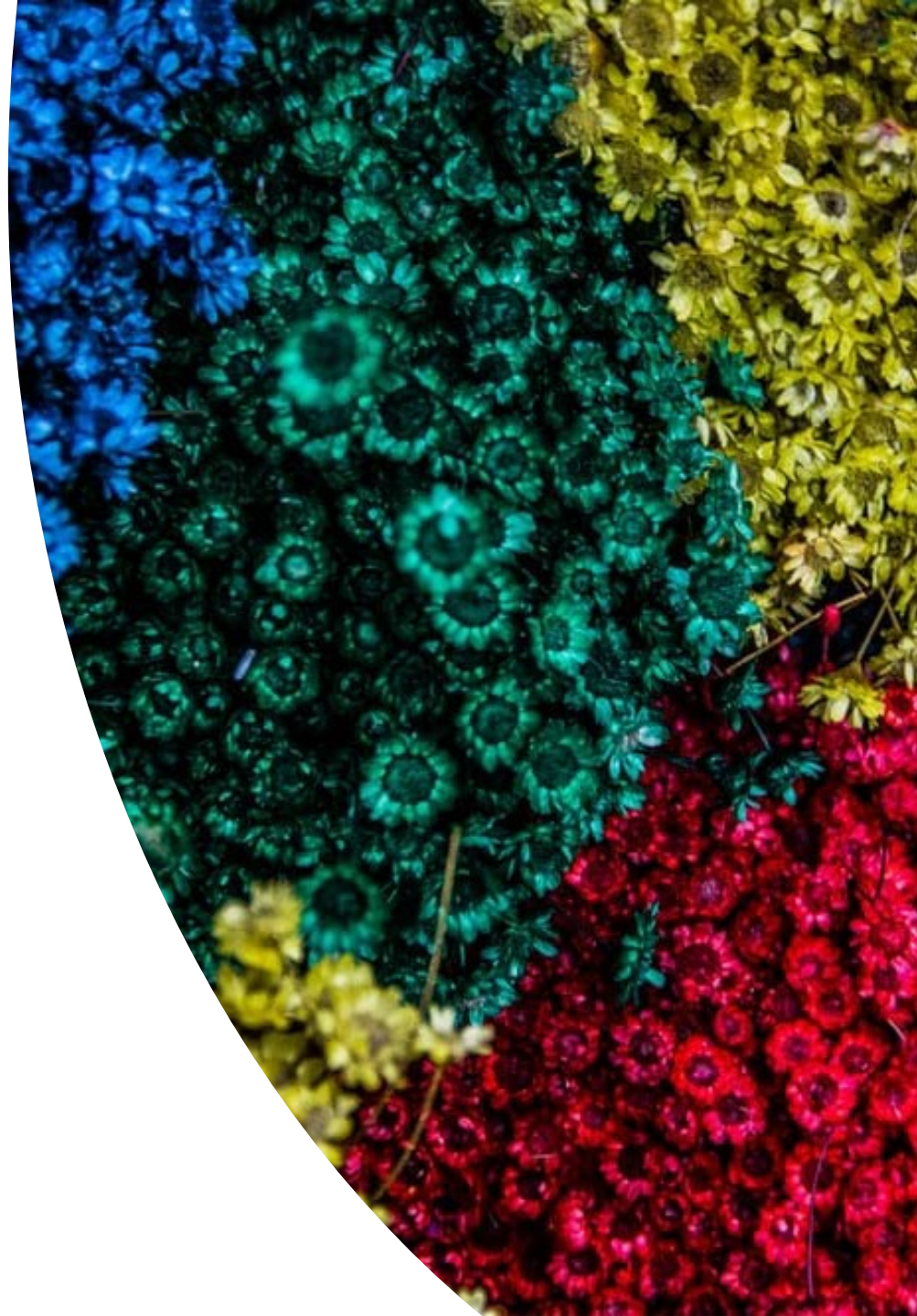
Use a larger  
number of clusters

Several clusters  
represent a true  
cluster

(c) Non-spherical shapes.

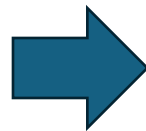
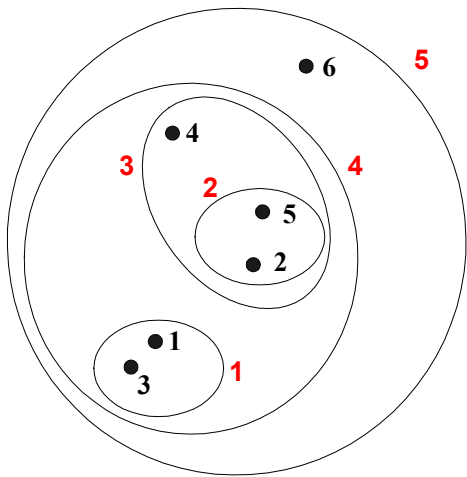
# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- **Clustering Algorithms**
  - K-Means Clustering
  - Hierarchical Clustering**
  - Density-based Clustering
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation
- Outliers and Scaling Issues

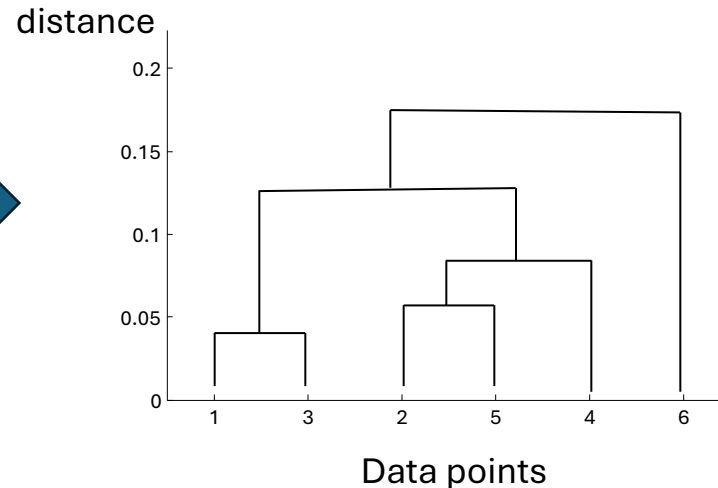


# Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree called a **dendrogram**. The dendrogram shows at what distance points join into a cluster.

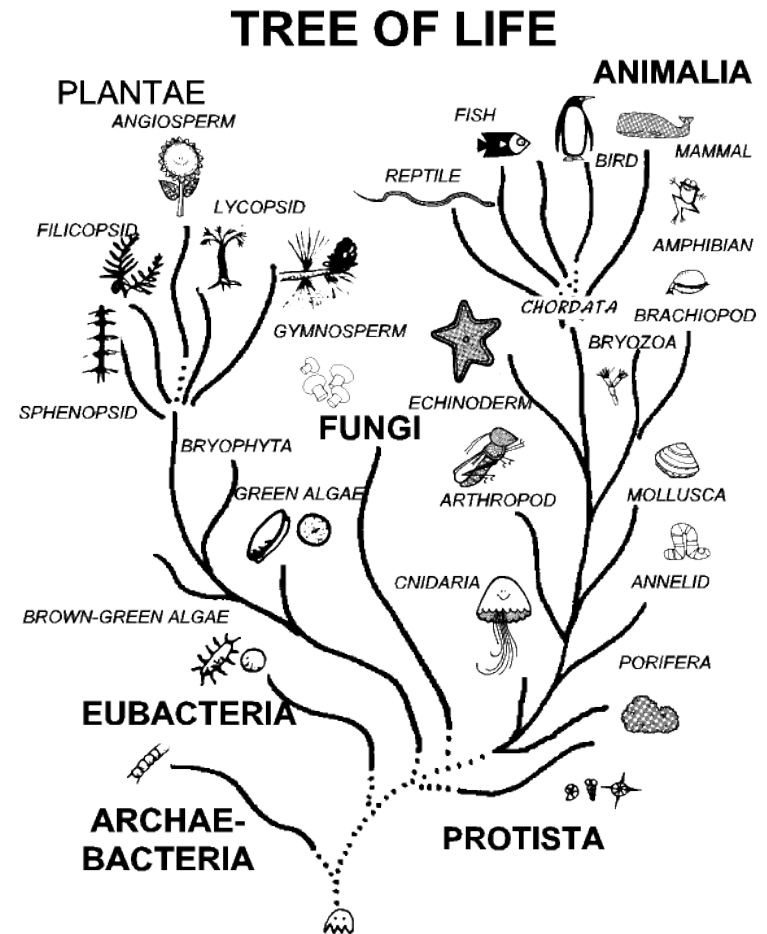


## Dendrogram



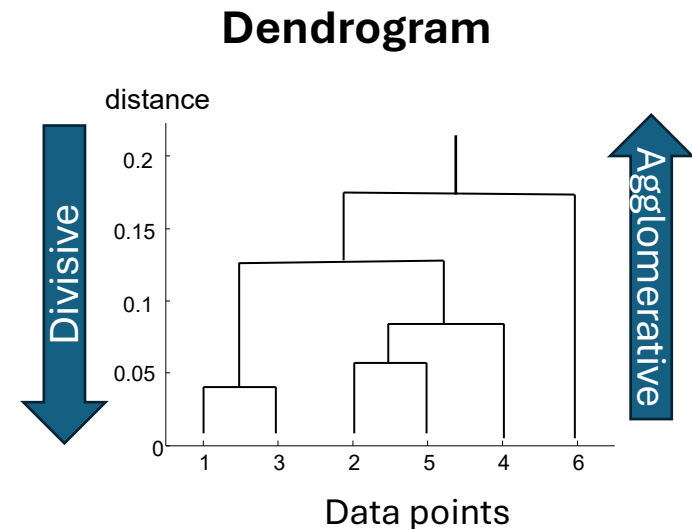
# Strengths of Hierarchical Clustering

- You do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level.
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)



# Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms
  - Use a similarity or distance matrix
  - Agglomerative: merge one cluster at a time





# Agglomerative Clustering Algorithm

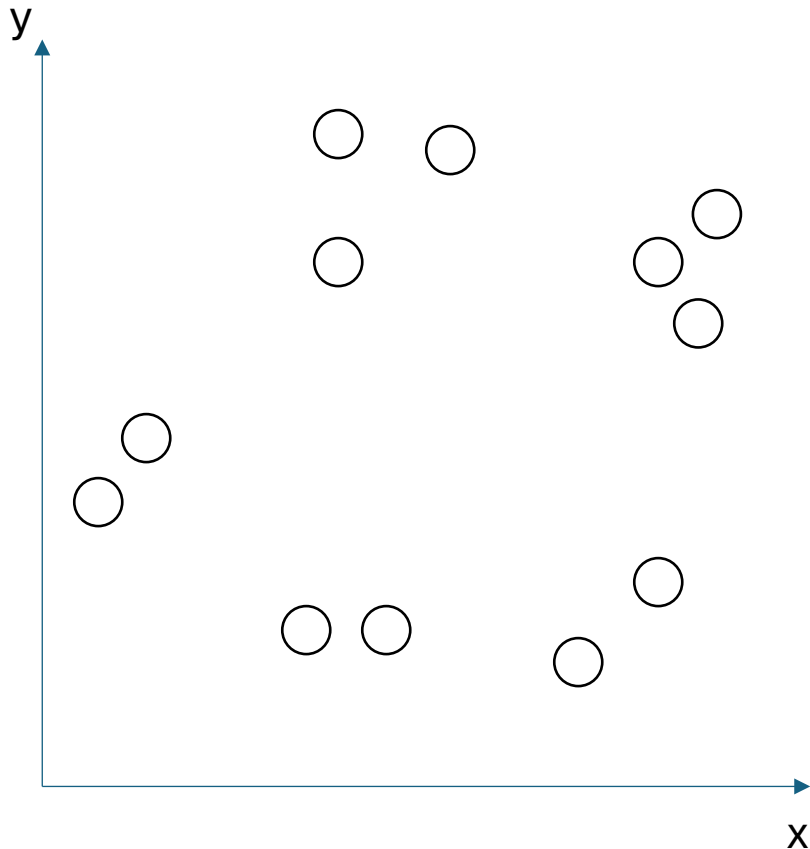
- The basic algorithm is straightforward

1. Compute the proximity matrix
2. Let each data point be a cluster
3. Repeat
4.     Merge the two **closest** clusters
5.     Update the proximity matrix
6. Until only a single cluster remains

- The key operation is **merging the two closest clusters** which requires the computation of the proximity between two clusters.

# Starting Situation

- Start with clusters of individual points and a proximity matrix



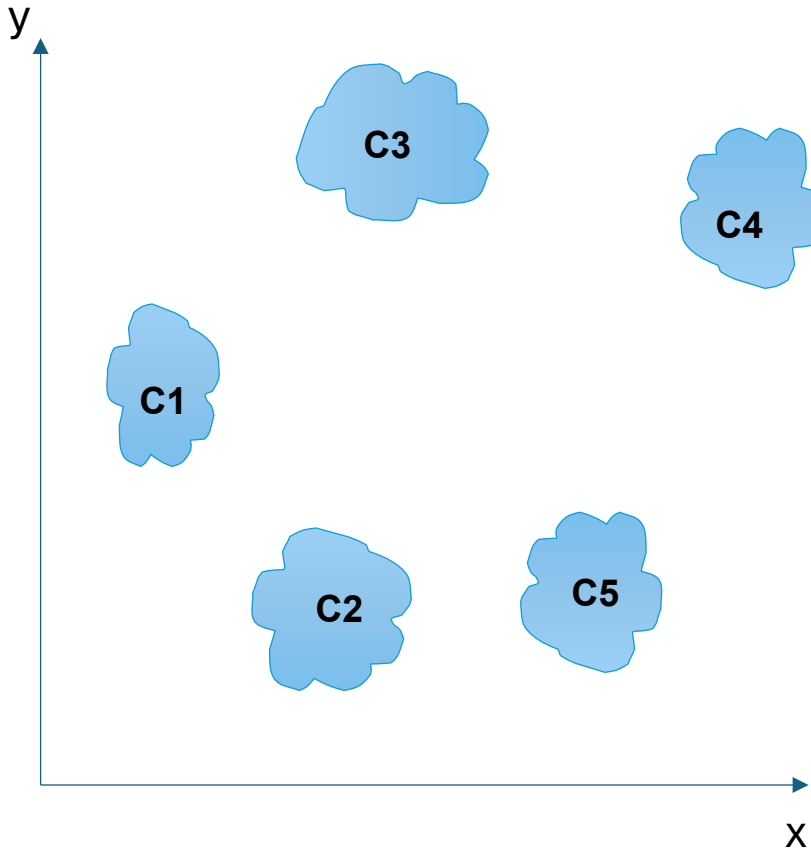
|    | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 |    |    |    |    |    |     |
| p2 |    |    |    |    |    |     |
| p3 |    |    |    |    |    |     |
| p4 |    |    |    |    |    |     |
| p5 |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |

**Proximity Matrix**



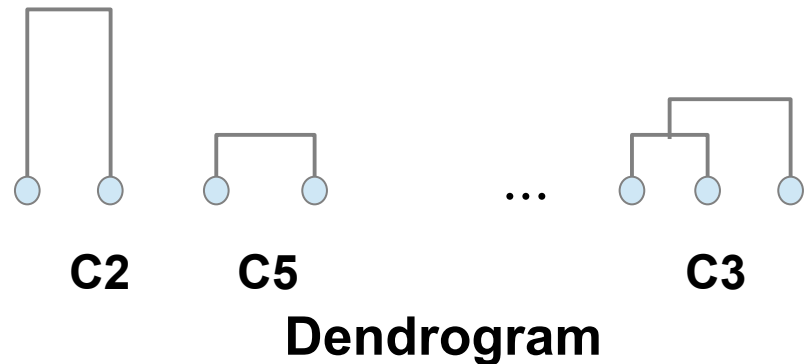
# Intermediate Situation

- After some merging steps, we have some clusters



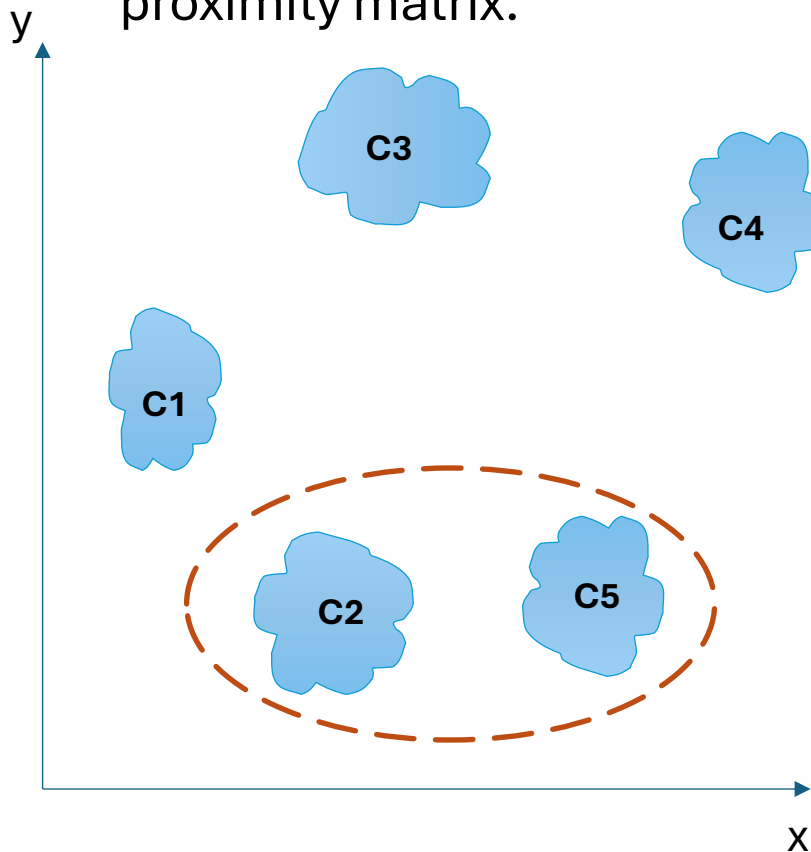
|    | C1 | C2 | C3 | C4 | C5 |
|----|----|----|----|----|----|
| C1 |    |    |    |    |    |
| C2 |    |    |    |    |    |
| C3 |    |    |    |    |    |
| C4 |    |    |    |    |    |
| C5 |    |    |    |    |    |

**Proximity Matrix**



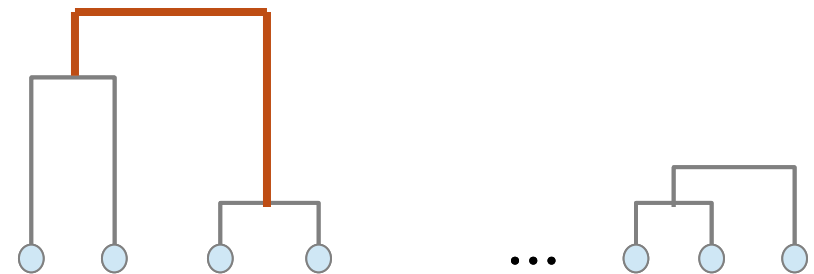
# Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



|    | C1 | C2 | C3 | C4 | C5 |
|----|----|----|----|----|----|
| C1 |    |    |    |    |    |
| C2 |    |    |    |    |    |
| C3 |    |    |    |    |    |
| C4 |    |    |    |    |    |
| C5 |    |    |    |    |    |

Proximity Matrix



C2

C5

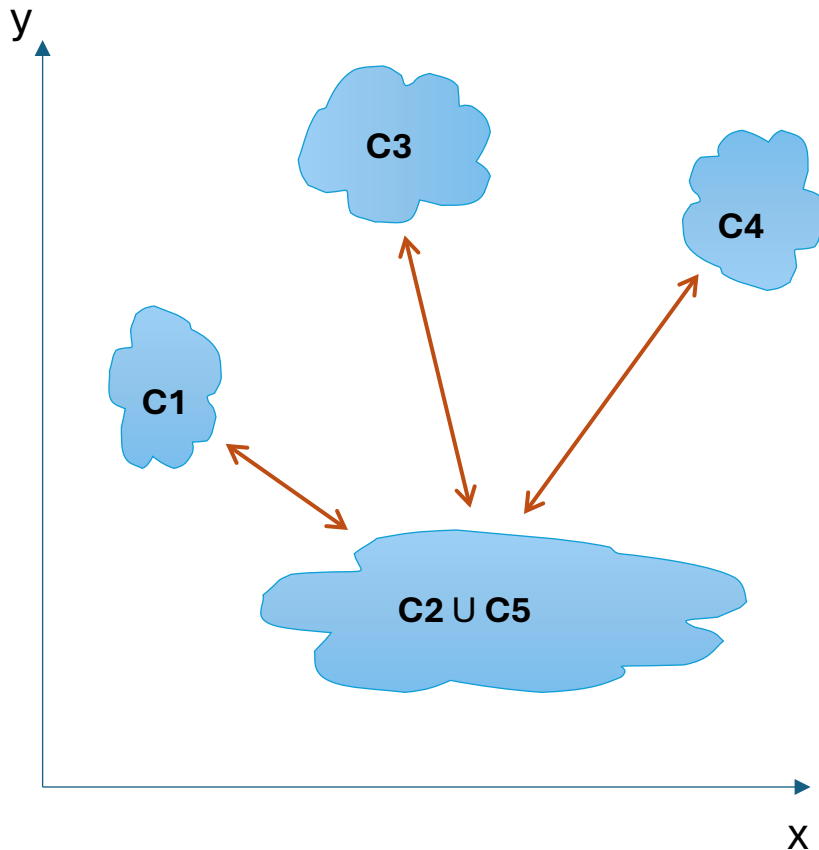
...

C3

Dendrogram

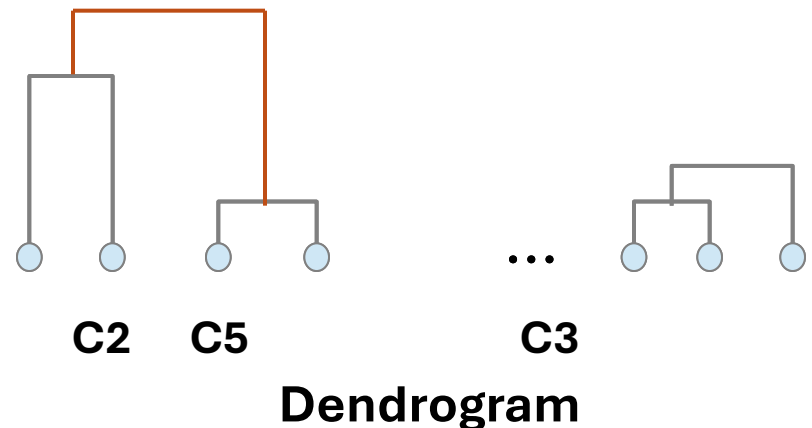
# After Merging

- The question is “How do we update the proximity matrix?”

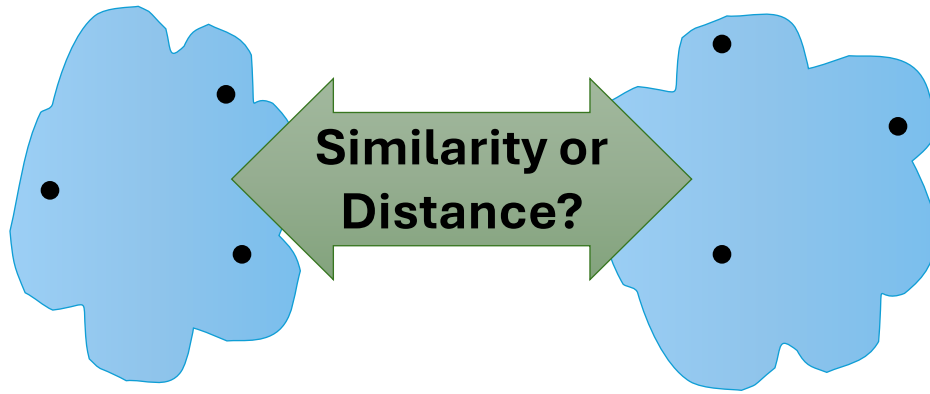


|         | C1 | C2<br>U<br>C5 | C3 | C4 |
|---------|----|---------------|----|----|
| C1      |    | ?             |    |    |
| C2 U C5 | ?  |               | ?  | ?  |
| C3      |    | ?             |    |    |
| C4      |    | ?             |    |    |

Proximity Matrix



# How to Define Inter-Cluster Similarity

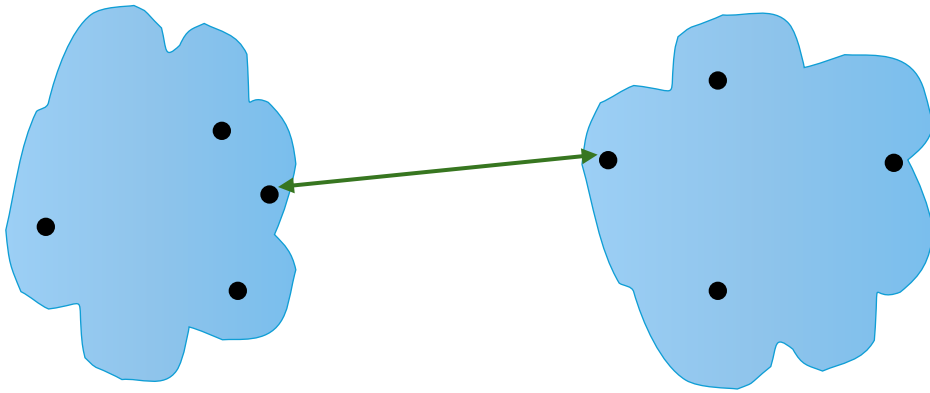


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

|    | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 |    |    |    |    |    |     |
| p2 |    |    |    |    |    |     |
| p3 |    |    |    |    |    |     |
| p4 |    |    |    |    |    |     |
| p5 |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |

• **Proximity Matrix**

# How to Define Inter-Cluster Similarity

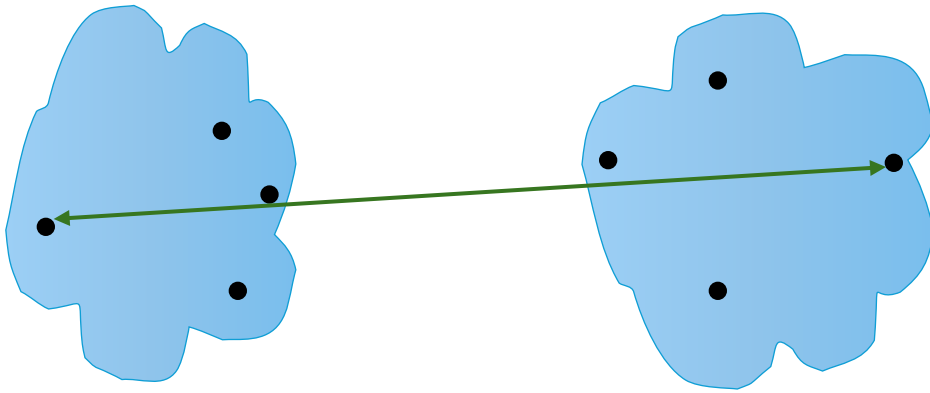


- **MIN (Single Link)**
- MAX (Complete Link)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

|    | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 |    |    |    |    |    |     |
| p2 |    |    |    |    |    |     |
| p3 |    |    |    |    |    |     |
| p4 |    |    |    |    |    |     |
| p5 |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |

Proximity Matrix

# How to Define Inter-Cluster Similarity



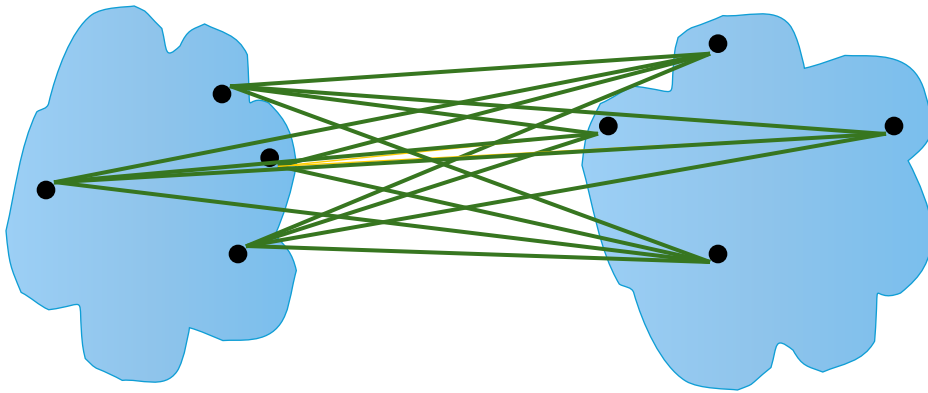
- MIN (Single Link)
- **MAX (Complete Link)**
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

|    | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 |    |    |    |    |    |     |
| p2 |    |    |    |    |    |     |
| p3 |    |    |    |    |    |     |
| p4 |    |    |    |    |    |     |
| p5 |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |

• **Proximity Matrix**



# How to Define Inter-Cluster Similarity

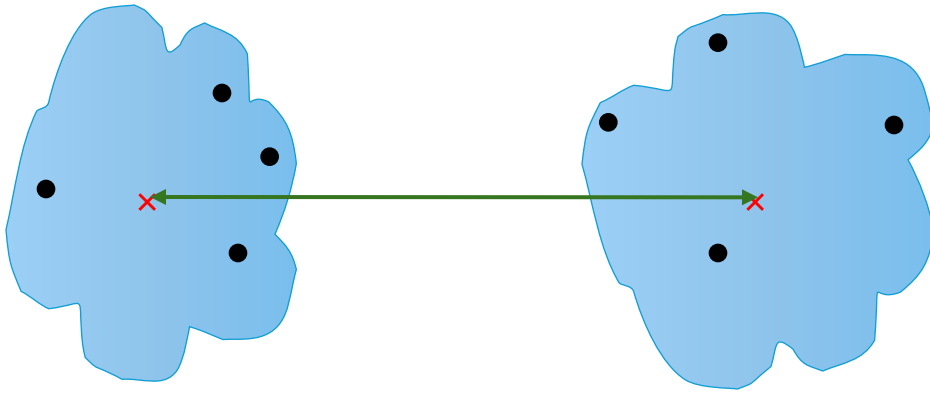


- MIN (Single Link)
- MAX (Complete Link)
- **Group Average (Average Link)**
- Distance Between Centroids
- Other methods driven by an objective function  
Ward's Method uses squared error

|    | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 |    |    |    |    |    |     |
| p2 |    |    |    |    |    |     |
| p3 |    |    |    |    |    |     |
| p4 |    |    |    |    |    |     |
| p5 |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |

• **Proximity Matrix**

# How to Define Inter-Cluster Similarity

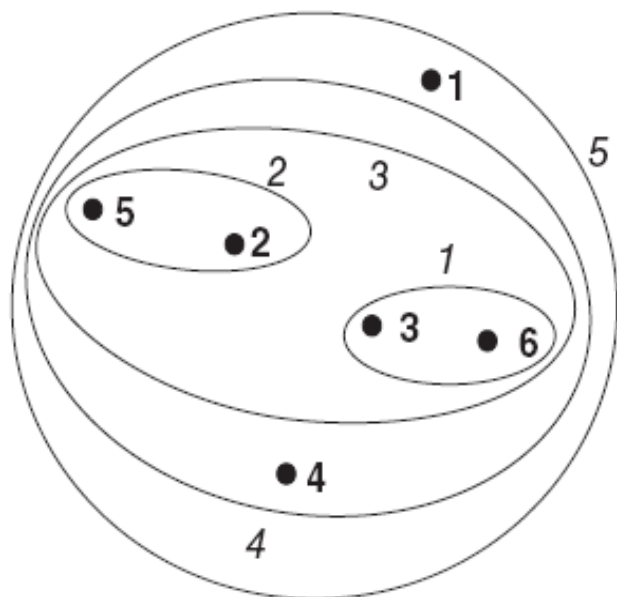


|    | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 |    |    |    |    |    |     |
| p2 |    |    |    |    |    |     |
| p3 |    |    |    |    |    |     |
| p4 |    |    |    |    |    |     |
| p5 |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |
| .  |    |    |    |    |    |     |

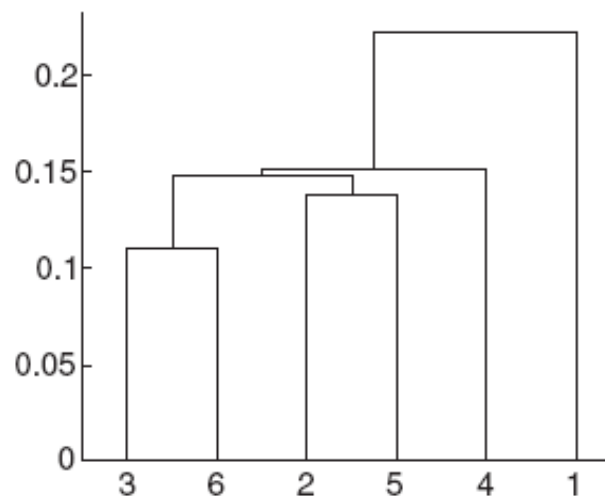
• **Proximity Matrix**

- MIN (Single Link)
- MAX (Complete Link)
- Group Average
- **Distance Between Centroids**
- Other methods driven by an objective function
  - Ward's Method uses squared error

# Single Link



(a) Single link clustering.

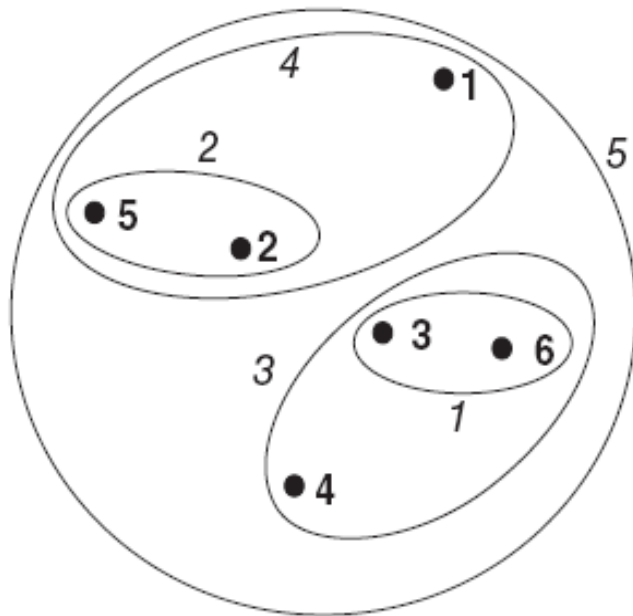


(b) Single link dendrogram.

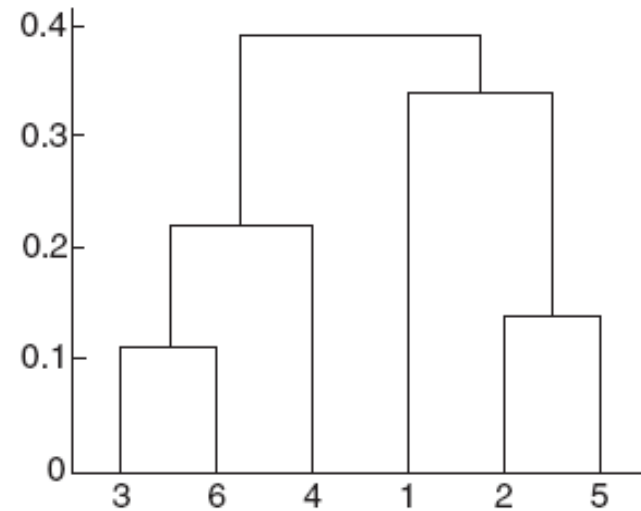
**Advantage:** Non-spherical, non-convex clusters

**Problem:** Chaining

# Complete Link



(a) Complete link clustering.

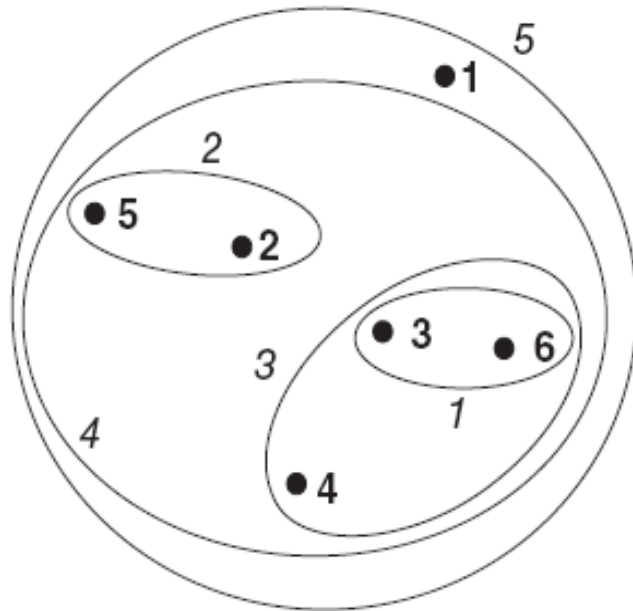


(b) Complete link dendrogram.

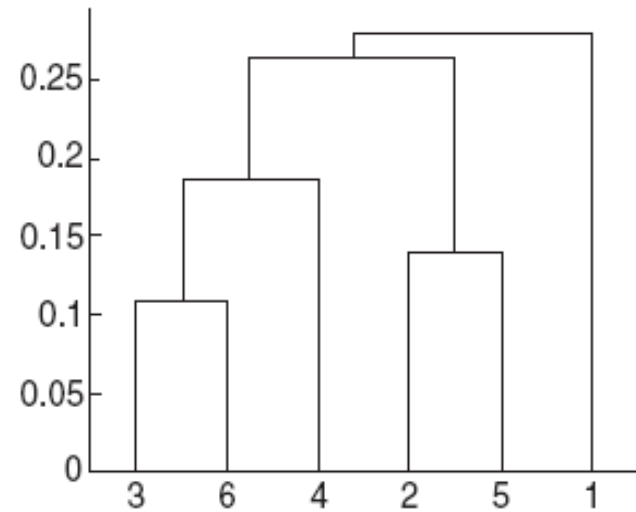
**Advantage:** more robust against noise (no chaining)

**Problem:** Tends to break large clusters,  
Biased towards globular clusters

# Average Link



(a) Group average clustering.



(b) Group average dendrogram.

A compromise between Single and Complete Link

# Ward's Method

- **Hierarchical analogue of K-means.**
- Similarity of two clusters is based on the increase in  $SSE$  when two clusters are merged.
- Less susceptible to noise and outliers.
- Biased towards globular clusters.

# Hierarchical Clustering: Complexity

- Space:  $O(N^2)$  since it uses the proximity matrix.
  - $N$  is the number of points.

This restricts the number of points that can be clustered!

- Time:  $O(N^3)$  in many cases
  - There are  $N - 1$  merge steps. At each step, the proximity matrix has to be searched and updated which takes  $O(N^2)$ .
  - Complexity can be reduced to  $O(N^2 \log(N))$  time for some approaches.
  - Single-link hierarchical clustering is the cheapest and can be done in  $O(N^2)$ .

# Hierarchical Clustering: Limitations

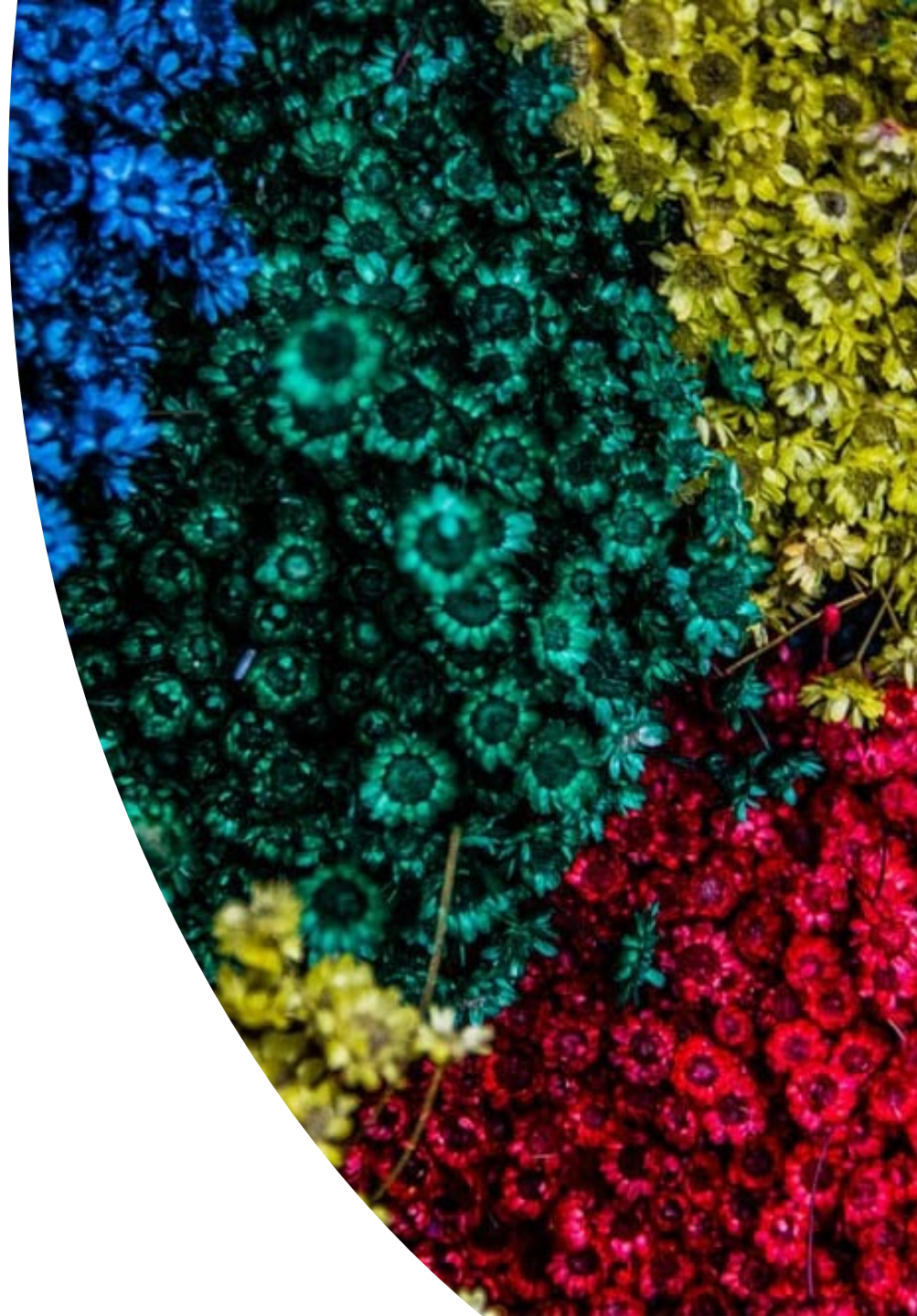
- **Greedy:** Once a decision is made to combine two clusters, it cannot be undone.
- **No global objective function,** just local merge decisions.
- Different schemes have problems with one or more of the following:
  - General: Sensitivity to noise and outliers (uses a complete clustering approach).
  - Complete –link: Difficulty handling different sized clusters (e.g., breaking large clusters apart) and convex shapes.
  - Single-link: Chaining.





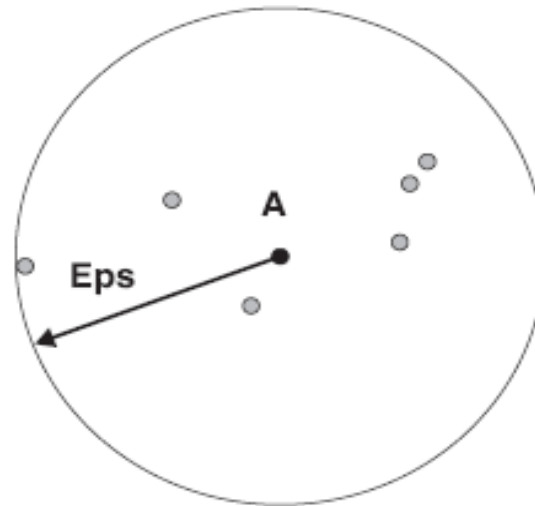
# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- **Clustering Algorithms**
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering**
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation
- Outliers and Scaling Issues



# Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

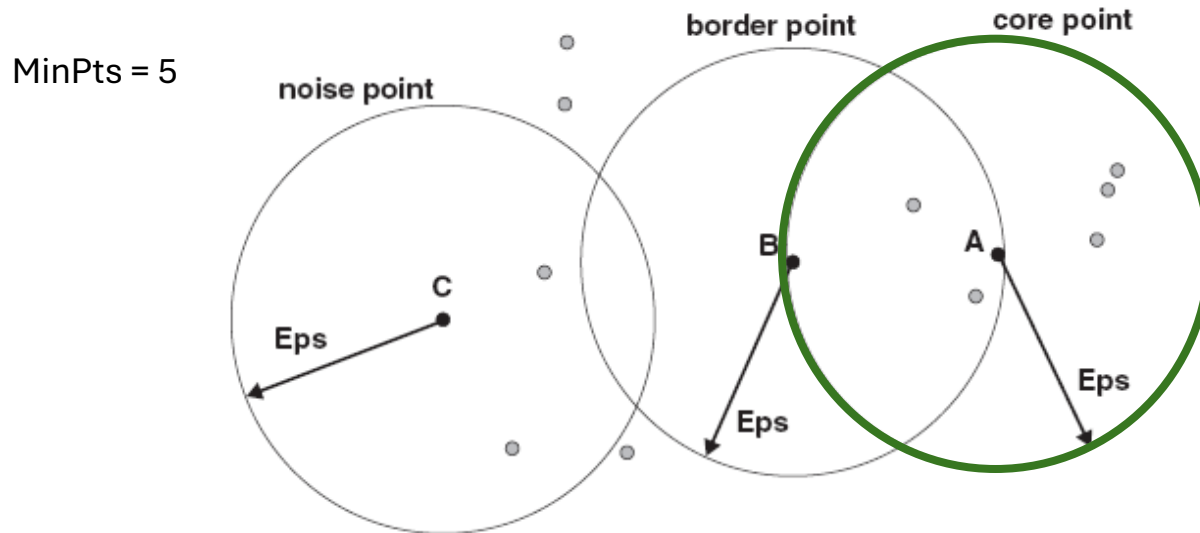
Measures density around a point as the number of points within a specified radius Eps called the eps-neighborhood.



density = 7 points

# DBSCAN

- Classifies points by the density of their eps-neighborhood:
  - point is a **core point** if it has more than a specified number of points (MinPts) within Eps. These are points that form the interior of a cluster.
  - A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A **noise point** is any point that is not a core point or a border point.

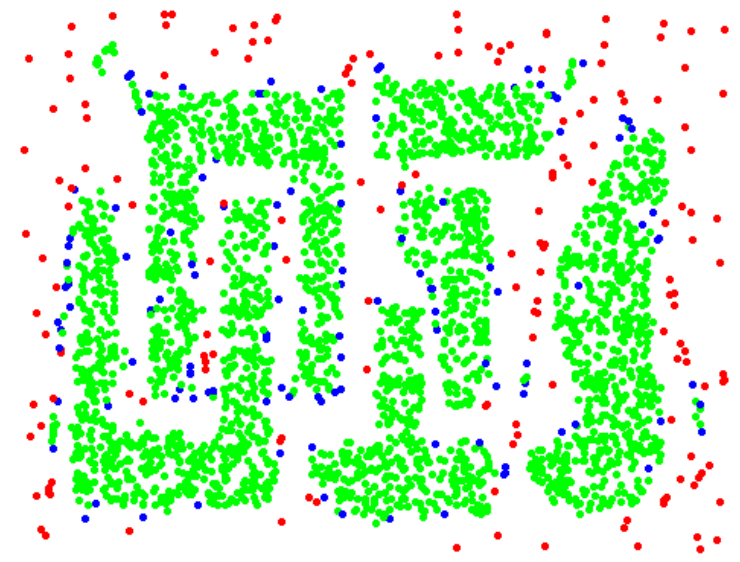


# DBSCAN: Core, Border and Noise Points



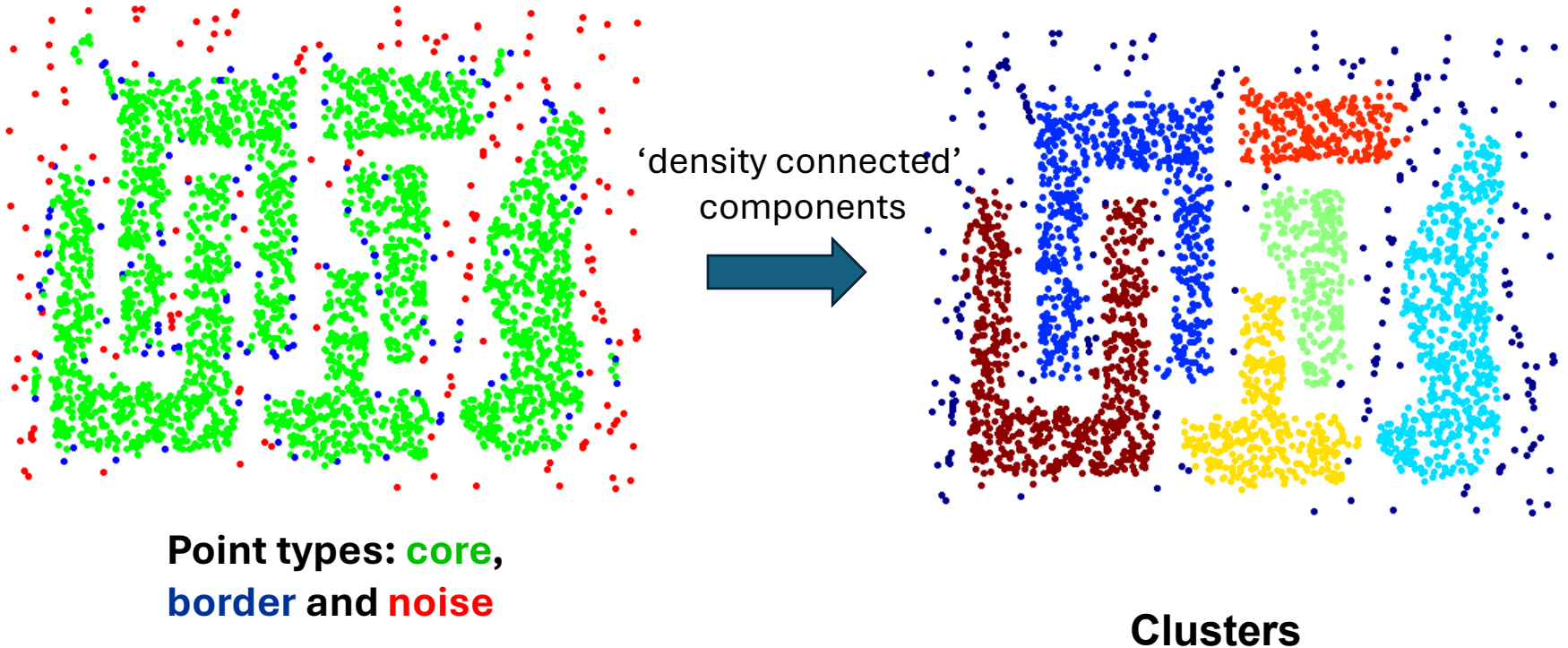
Original Points

Eps = 10  
MinPts = 4



Point types: **core**,  
**border** and **noise**

# DBSCAN: Determine Clusters



## Properties of DBSCAN

- Resistant to **noise and outliers**.
- Can handle clusters of different **shapes and sizes**.
- **Eps and MinPts** depend on each other and can be hard to specify.

# DBSCAN Algorithm

**DBSCAN**(D, eps, MinPts)

C = 0

for each unvisited point P in dataset D

mark P as visited

NeighborPts = regionQuery(P, eps)

if sizeof(NeighborPts) < MinPts

mark P as NOISE

else

C = next cluster

expandCluster(P, NeighborPts, C, eps, MinPts)

**expandCluster**(P, NeighborPts, C, eps, MinPts)

add P to cluster C

for each point P' in NeighborPts

if P' is not visited

mark P' as visited

NeighborPts' = regionQuery(P', eps)

if sizeof(NeighborPts') >= MinPts

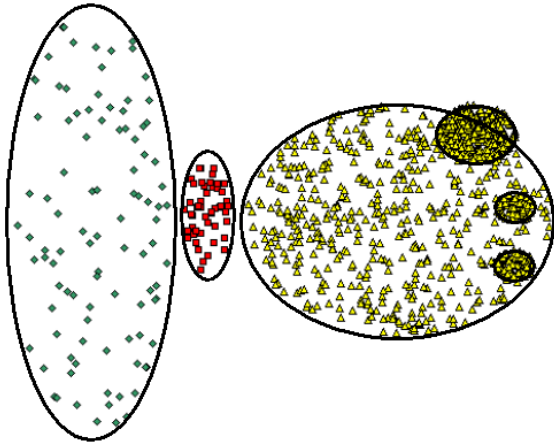
NeighborPts = NeighborPts joined with NeighborPts'

if P' is not yet member of any cluster

add P' to cluster C

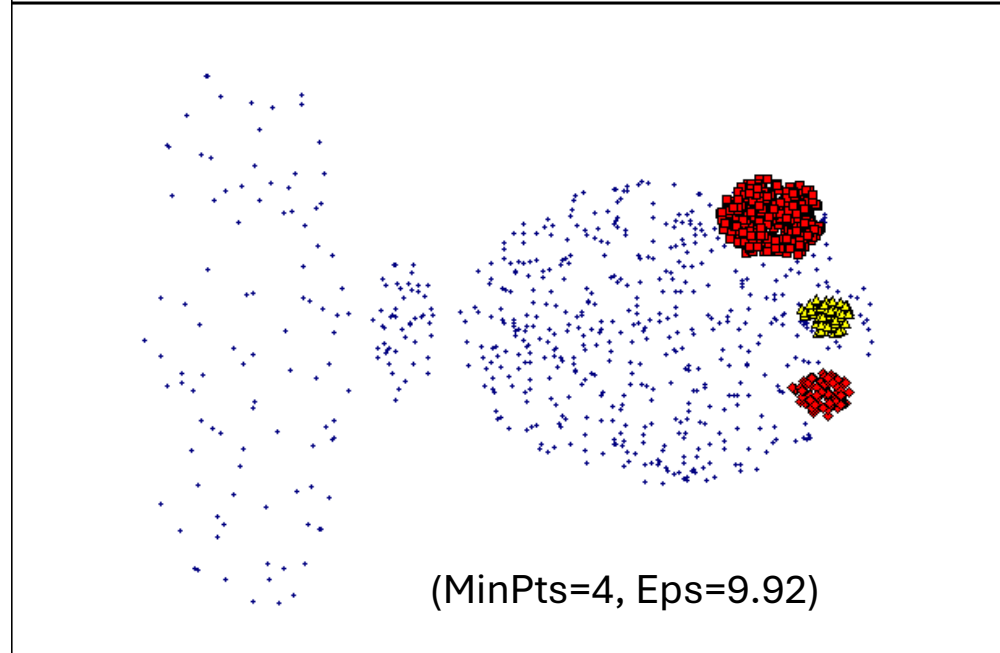
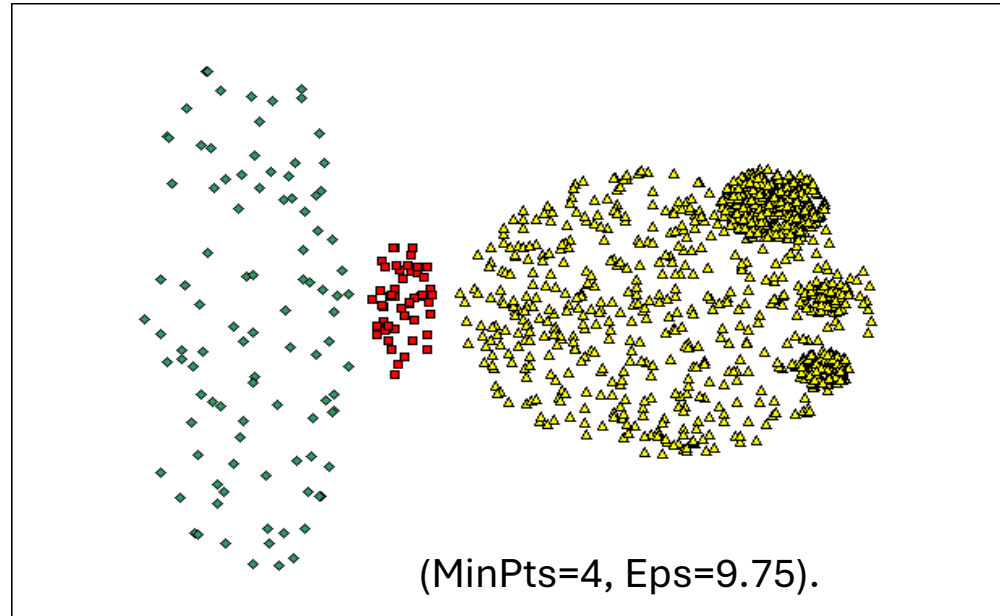
**Note:** Finding neighbors using region queries is the most expensive operation.

# When DBSCAN Does NOT Work Well



**Original Points**

- Varying densities
- High-dimensional data

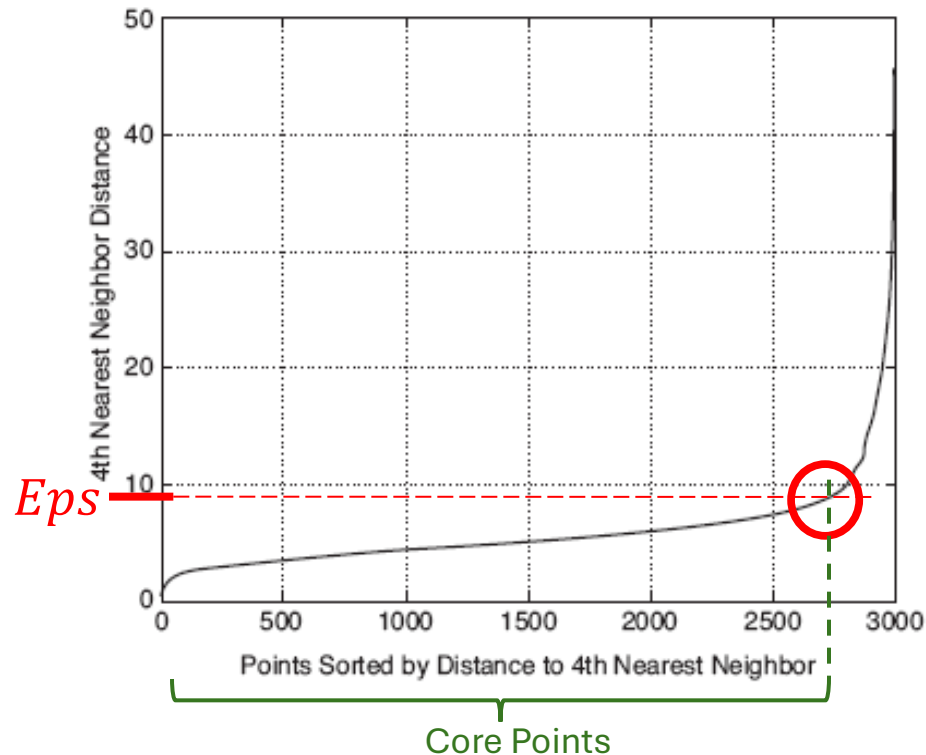


# DBSCAN: Determining EPS and MinPts

- Idea:

- Points in a cluster (core points) are close to other points and have a small  $k$ -nearest neighbor distance.
- Noise points are in a low-density area and have a larger  $k$ -nearest neighbor distance.

- Plot all points sorted by their  $k$ -nearest neighbor distance and find the knee where the  $k$ -nearest neighbor distance starts to increase fast.



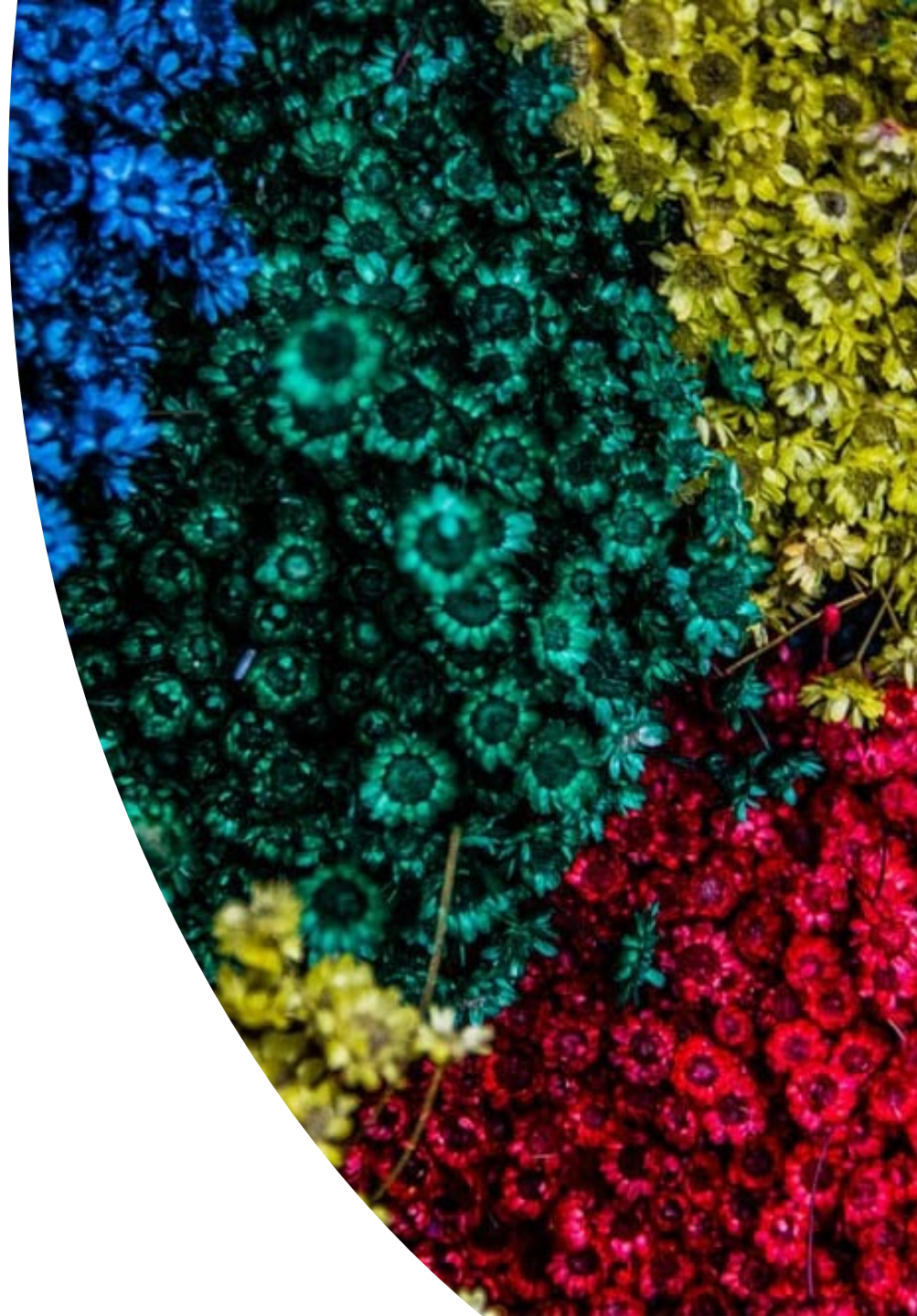
$$k = \text{MinPts} - 1$$





# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
  - Other Methods**
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation
- Outliers and Scaling Issues



# Some Other Clustering Algorithms

## ■ Center-based Clustering

- Fuzzy c-means
- PAM (Partitioning Around Medoids)

## ■ Mixture Models

- Expectation-maximization (EM) algorithm

## ■ Hierarchical

- CURE (Clustering Using Representatives): shrinks points toward center
- BIRCH (balanced iterative reducing and clustering using hierarchies)

## ■ Graph-based Clustering

- Graph partitioning on a sparsified proximity graph
- Shared nearest-neighbor (SNN graph)

## ■ Spectral Clustering

- Reduce the dimensionality using the spectrum of the similarity, and cluster in this space.

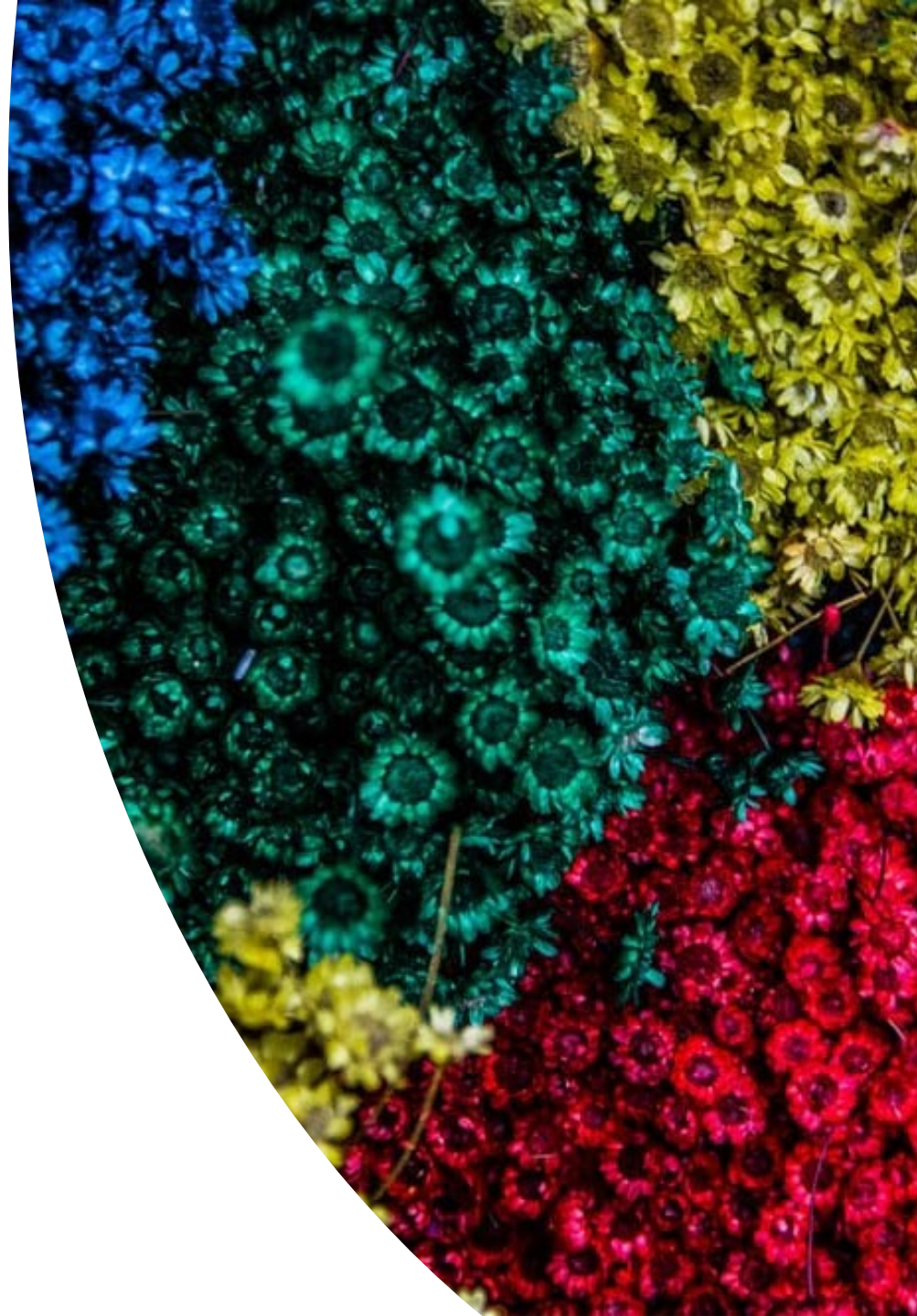
## ■ Subspace Clustering

## ■ Data Stream Clustering



# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- **Cluster Evaluation**
  - Unsupervised Evaluation
  - Supervised Evaluation
- Outliers and Scaling Issues



# Cluster Evaluation

- For supervised classification (= we have a class label) we have a variety of measures to evaluate how good our model is:

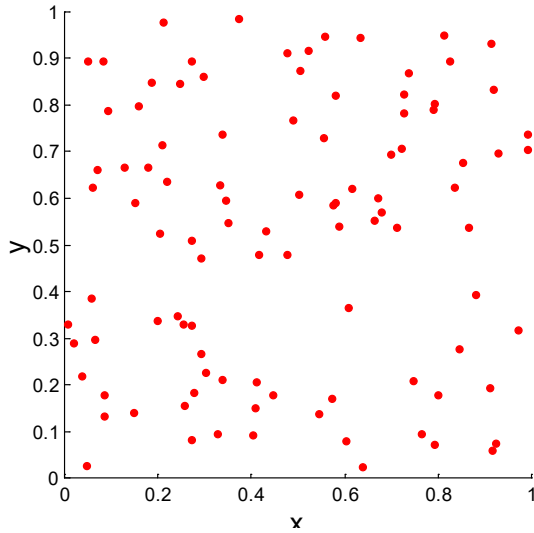
Accuracy, error rate, loss, precision, recall,...

- For cluster analysis (=unsupervised learning), the analogous question is:

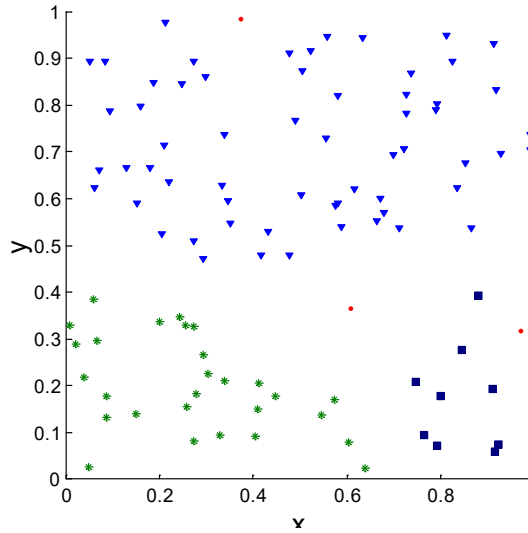
**How to evaluate the “goodness” of the resulting clusters?**

# Clusters found in Random Data (Overfitting)

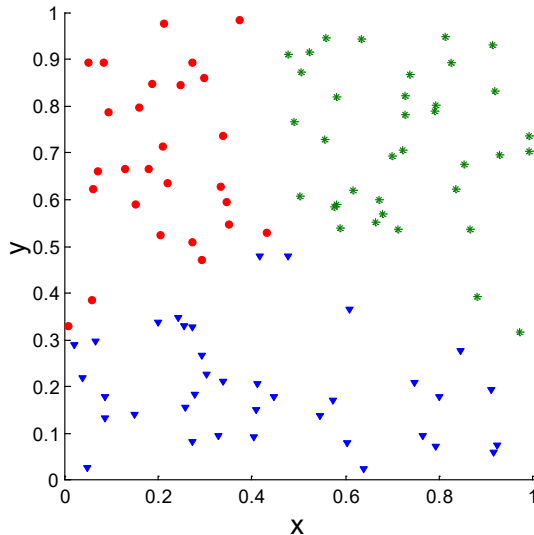
Random  
Points



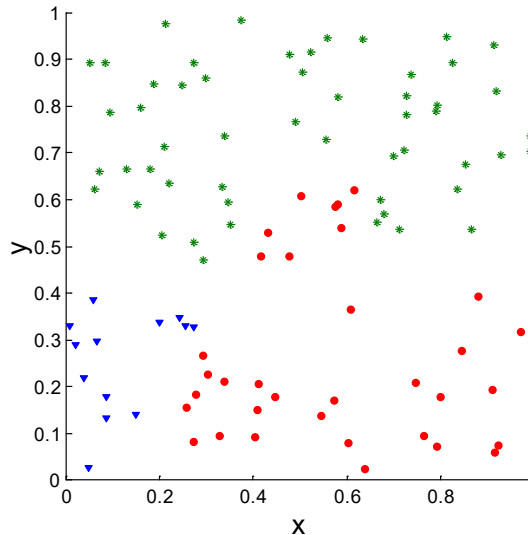
DBSCAN



K-means  
k=3



Complete Link  
+ cut into 3  
clusters



**If you tell a clustering algorithm to find clusters then it will!**

# Different Aspects of Cluster Evaluation

- 1. Unsupervised Evaluation:** Evaluate how well the results of a cluster analysis fit the data without reference to external group information.
  - Determining the **clustering tendency** of a set of data, i.e., distinguishing whether a non-random structure actually exists in the data (e.g., to avoid overfitting).
  - Determining the **‘correct’ number of clusters**.
- 2. Supervised Evaluation:** Compare the results of a cluster analysis to externally known group labels (ground truth).
- 3. Compare different clusterings** to determine which one is better.

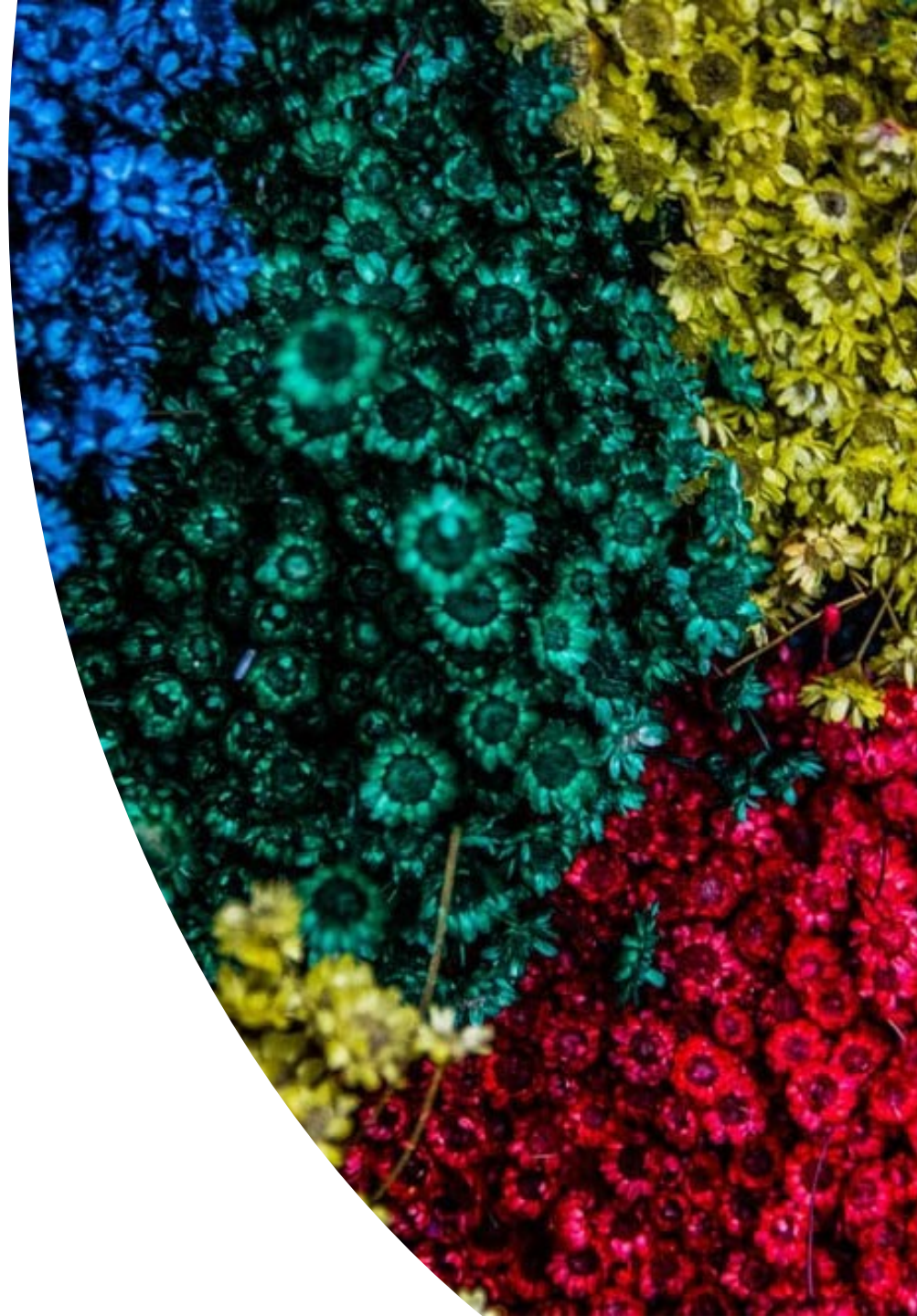
# Measures for Cluster Evaluation

Numerical measures that are applied to judge various aspects of cluster quality, are classified into the following three types.

- **Internal Index (unsupervised evaluation):** Used to measure the goodness of a clustering structure without respect to external information.
  - E.g.: Sum of Squared Error (SSE), Silhouette coefficient, Correlation between proximity and incidence matrix
- **External Index (supervised evaluation):** Used to measure the extent to which cluster labels match externally supplied group labels.
  - E.g., Entropy, Purity, Rand index, ...
- **Relative Index:** Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Evaluation
  - Unsupervised Evaluation**
  - Supervised Evaluation
- Outliers and Scaling Issues





# Measures for Cluster Evaluation

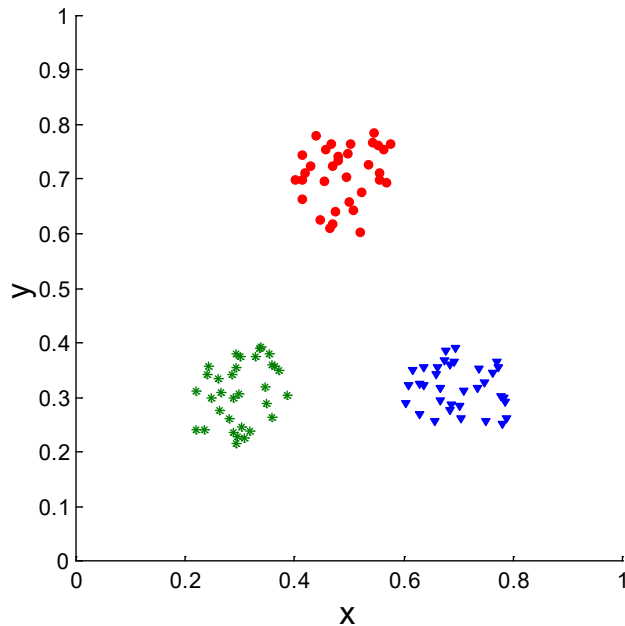
Numerical measures that are applied to judge various aspects of cluster quality, are classified into the following three types.

- **Internal Index (unsupervised evaluation):** Used to measure the goodness of a clustering structure without respect to external information.
  - E.g.: Sum of Squared Error (SSE), Silhouette coefficient
- **External Index (supervised evaluation):** Used to measure the extent to which cluster labels match externally supplied group labels.
  - E.g., Entropy, Purity, Rand index, ...
- **Relative Index:** Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

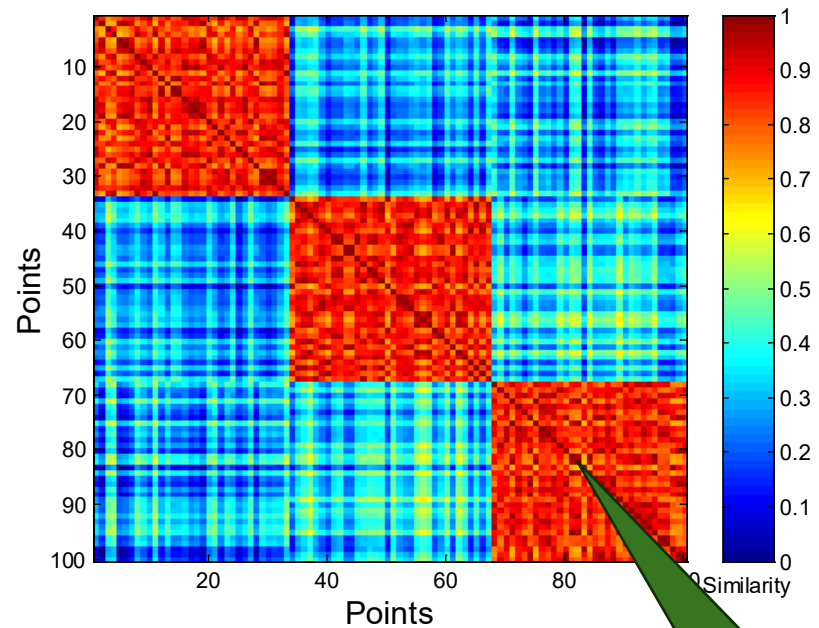
# Visual Method: Similarity Matrix Visualization

- Order the similarity matrix with respect to cluster labels and inspect visually.

Clustered Data



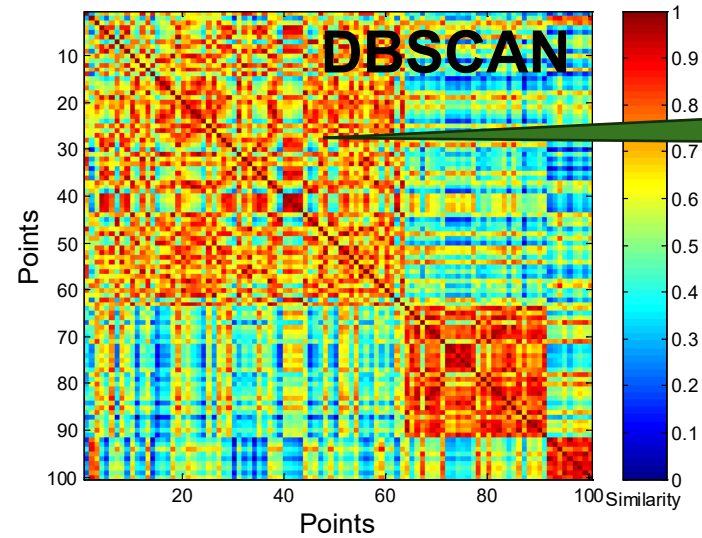
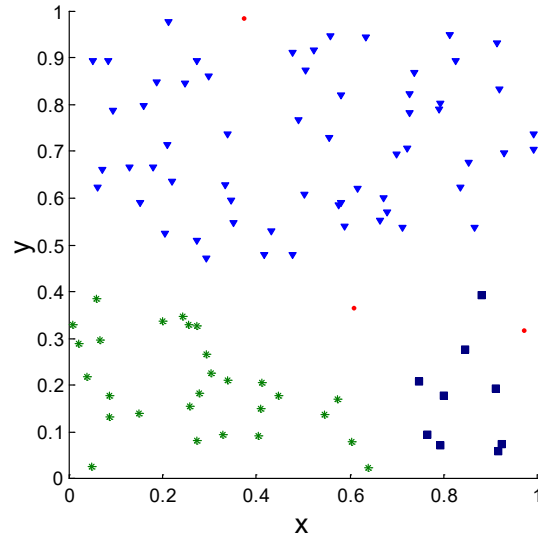
Reordered Similarity Matrix



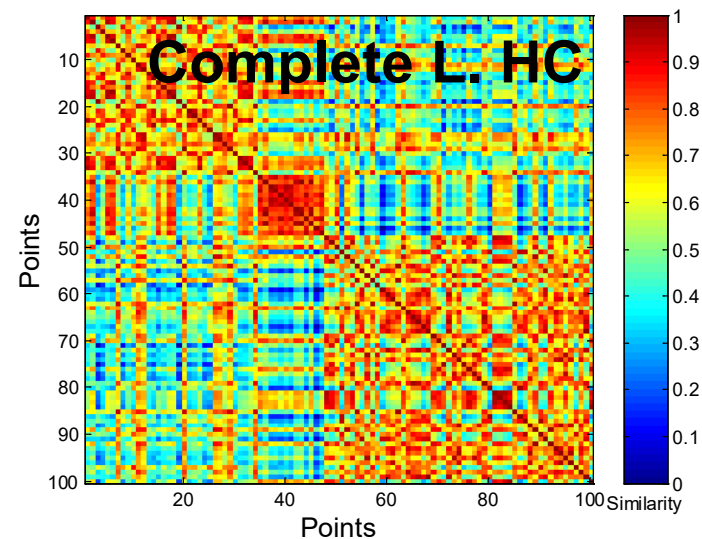
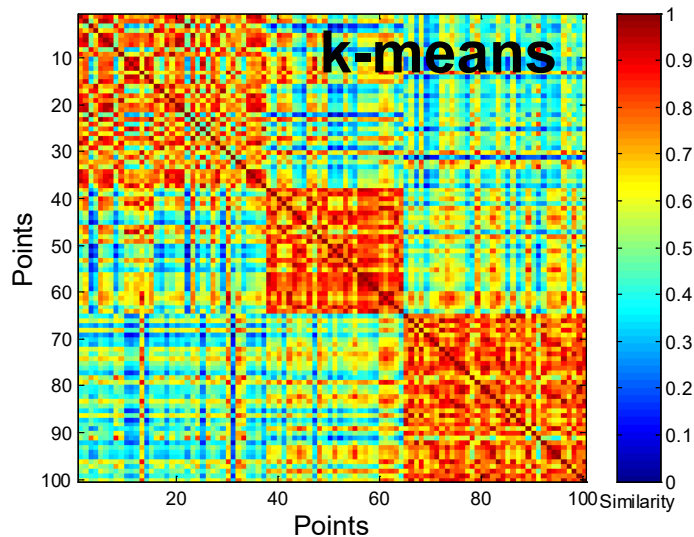
Clusters appear as crisp squares

# Visual Method: Similarity Matrix Visualization

Random data



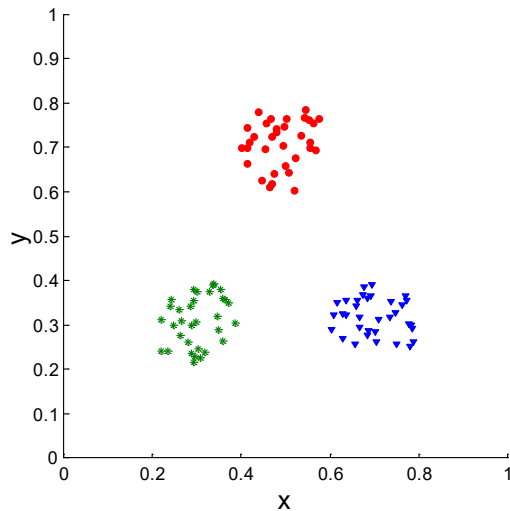
No crisp clusters!



# Correlation between Distances and Incidence Matrix

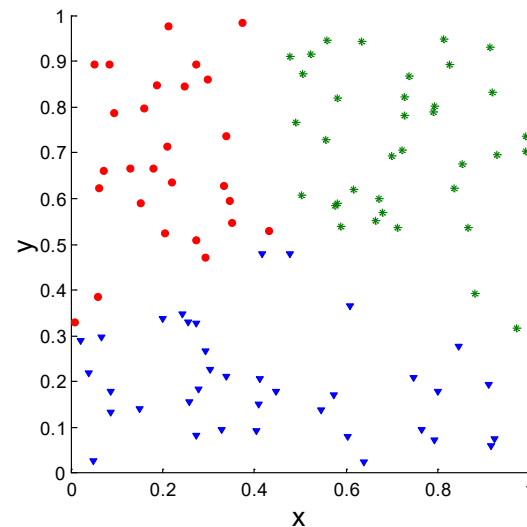
- Compute the correlation between the entries in
  - $n \times n$  similarity matrix representing the data
  - $n \times n$  incidence matrix representing the cluster memberships. A 1 in a row means that the points are in the same cluster.
- High correlation indicates that points that belong to the same cluster (a 1 in the incidence matrix) are close to each other (a large value in the similarity matrix).

**Good Cluster Structure**



**Corr = 0.9235**

**Random Data**

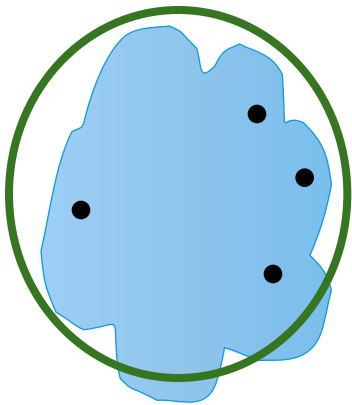


**Corr = 0.5810**

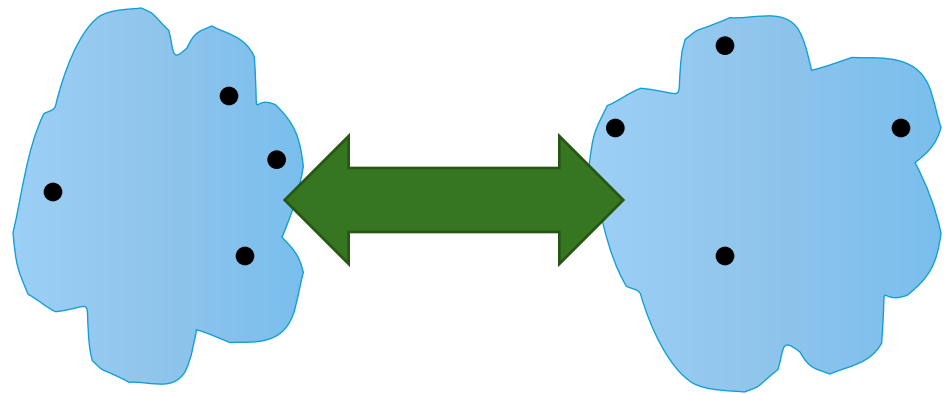
# Internal Index: Cohesion and Separation

Several indices are based on the concept of cohesion and separation.

- **Cluster Cohesion:** Measures how closely related objects in a cluster are.
- **Cluster Separation:** Measure how distinct or well-separated a cluster is from other clusters.



cohesion

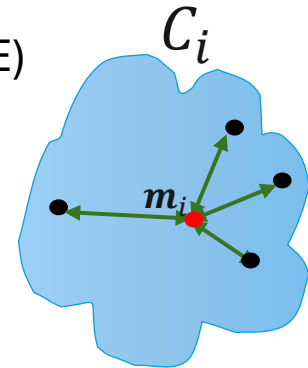


separation

# Internal Index: Sum of Squared Errors

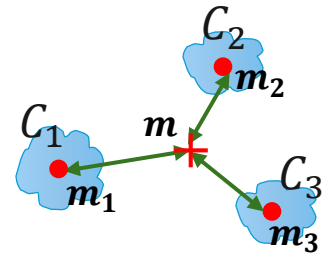
- **Cluster Cohesion:** Within cluster sum of squares (WSS = SSE)

$$WSS = \sum_{i=1}^K \sum_{x \in C_i} \|x - m_i\|^2$$



- **Cluster Separation:** Between cluster sum of squares (BSS)

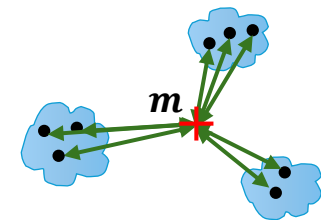
$$BSS = \sum_{i=1}^K |C_i| \|m_i - m\|^2$$



Where  $|C_i|$  is the size of cluster  $i$  and  $m$  is the centroid of the data space

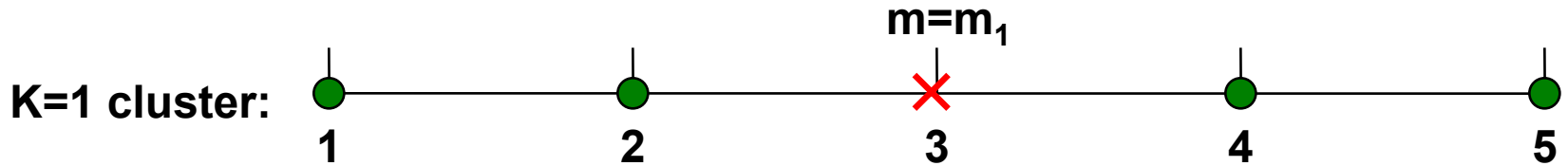
- Total sum of squares:  $TSS = \sum_{i=1}^n \|x_i - m\|^2$

$$TSS = WSS + BSS$$



# Internal Index: Sum of Squared Errors

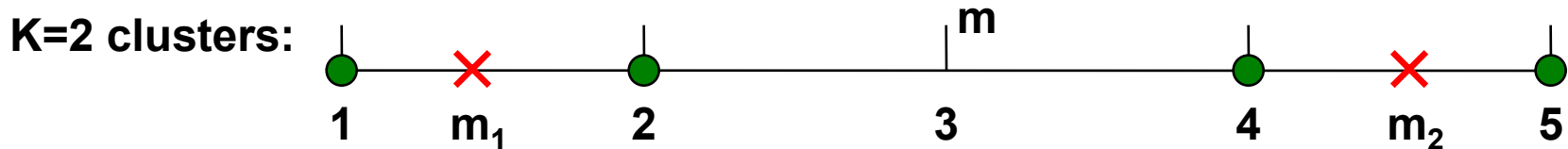
TSS = BSS + WSS = constant for a given data set



$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$



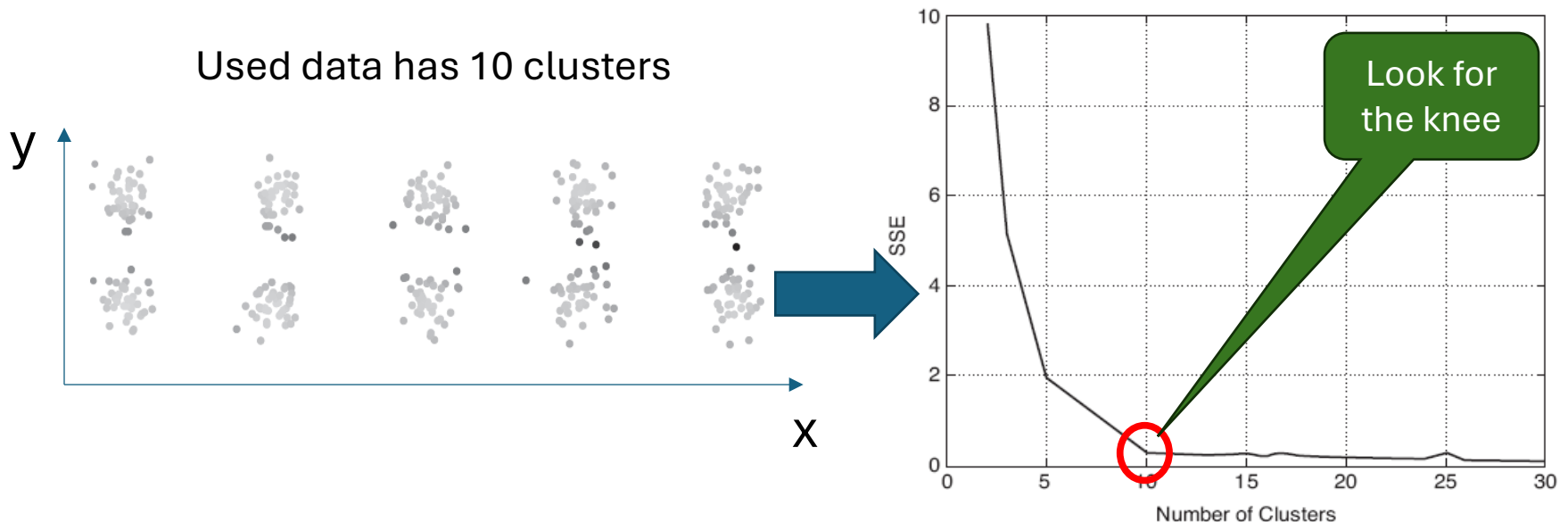
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

# Choosing k with the Sum of Squared Errors

- The SSE is good for comparing **two different clusterings** with the same number of clusters. E.g., several random restarts of k-means.
- We cannot directly compare the SSE between clusterings with different k. As k goes up, SSE tends to go down.
- We can also be used to estimate the **number of clusters using** the “knee” method.

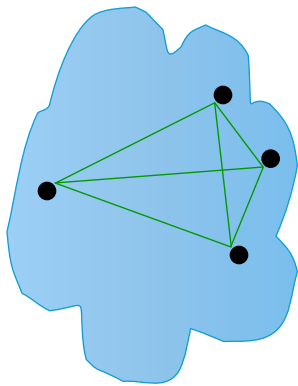




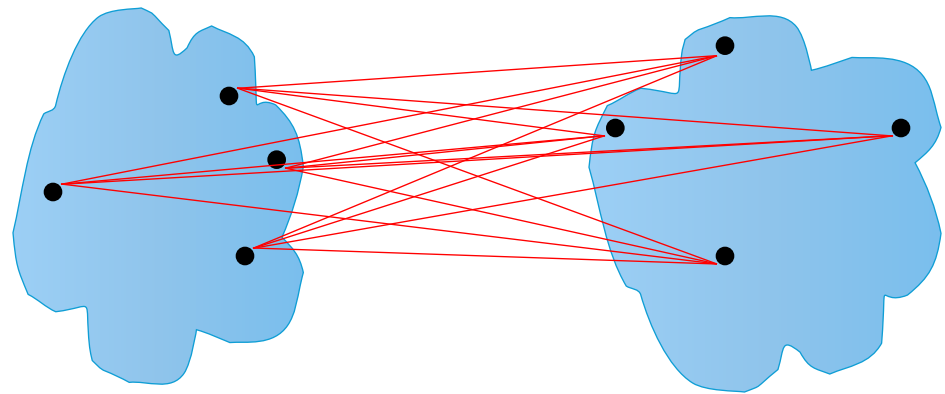
# Similarity-Graph Based Internal Index

A proximity graph-based approach can also be used for cohesion and separation. The edges in the graph represent similarities.

- **Cluster cohesion** is the sum of the similarities of all links within a cluster.
- **Cluster separation** is the sum of the similarities between nodes in the cluster and nodes outside the cluster.



cohesion



separation

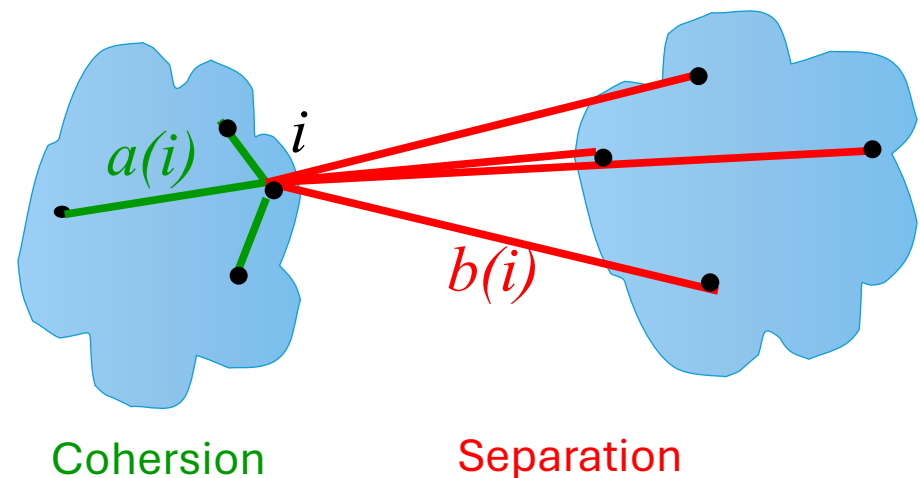
# Internal Index: Silhouette Coefficient

- The silhouette applies the similarity-graph based index to individual points. For an individual point  $i$ :
  - Calculate  $a(i)$  = average dissimilarity of  $i$  to all other points in its cluster
  - Calculate  $b(i)$  = lowest average dissimilarity of  $i$  to any other)

The silhouette index for point  $i$  is defined as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

Range:  $-1 \leq s(i) \leq 1$



- The closer to 1, the better.
- We can calculate the Average Silhouette Width (ASW) for a cluster or a clustering.

# Internal Index: Silhouette Plot

Silhouette plot of pam(x = dis.bc, k = 5)

n = 160

5 clusters  $C_j$

$j : n_j \mid \text{ave}_{i \in C_j} s_i$

1 : 32 | 0.20

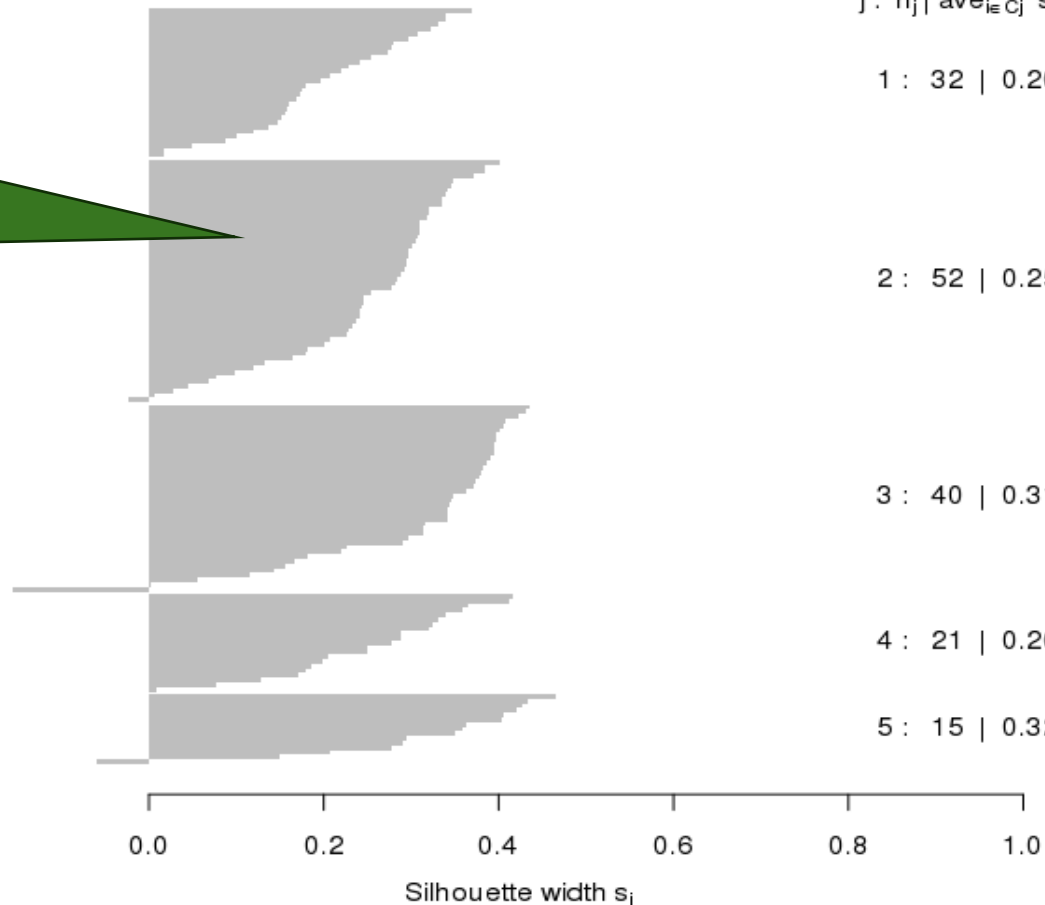
2 : 52 | 0.25

3 : 40 | 0.31

4 : 21 | 0.26

5 : 15 | 0.32

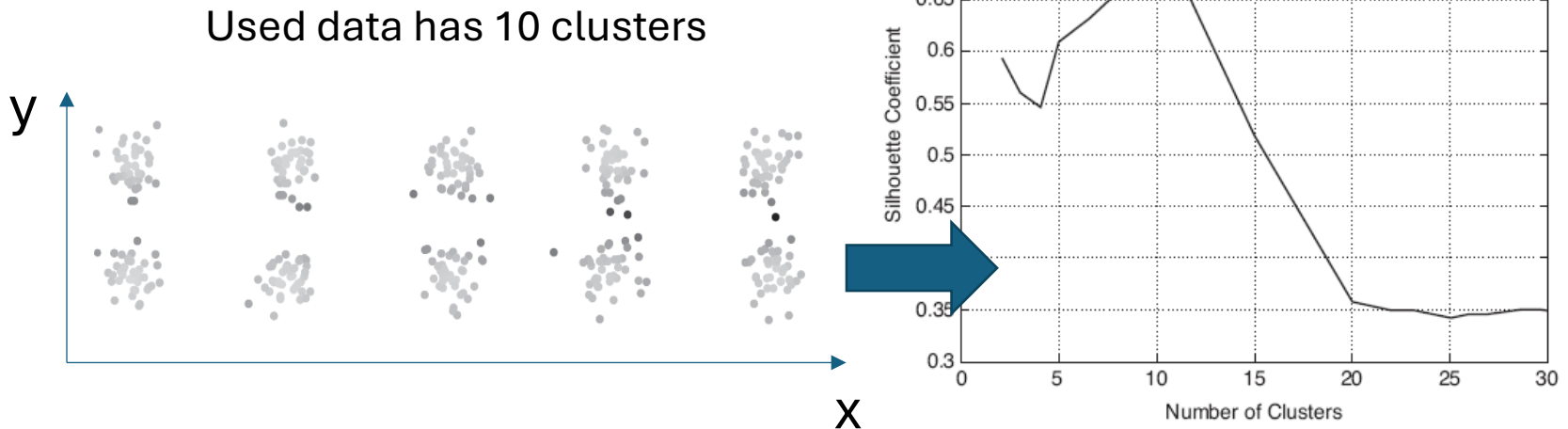
Large, positive  
silhouettes  
represent  
good clusters



Average silhouette width : 0.26

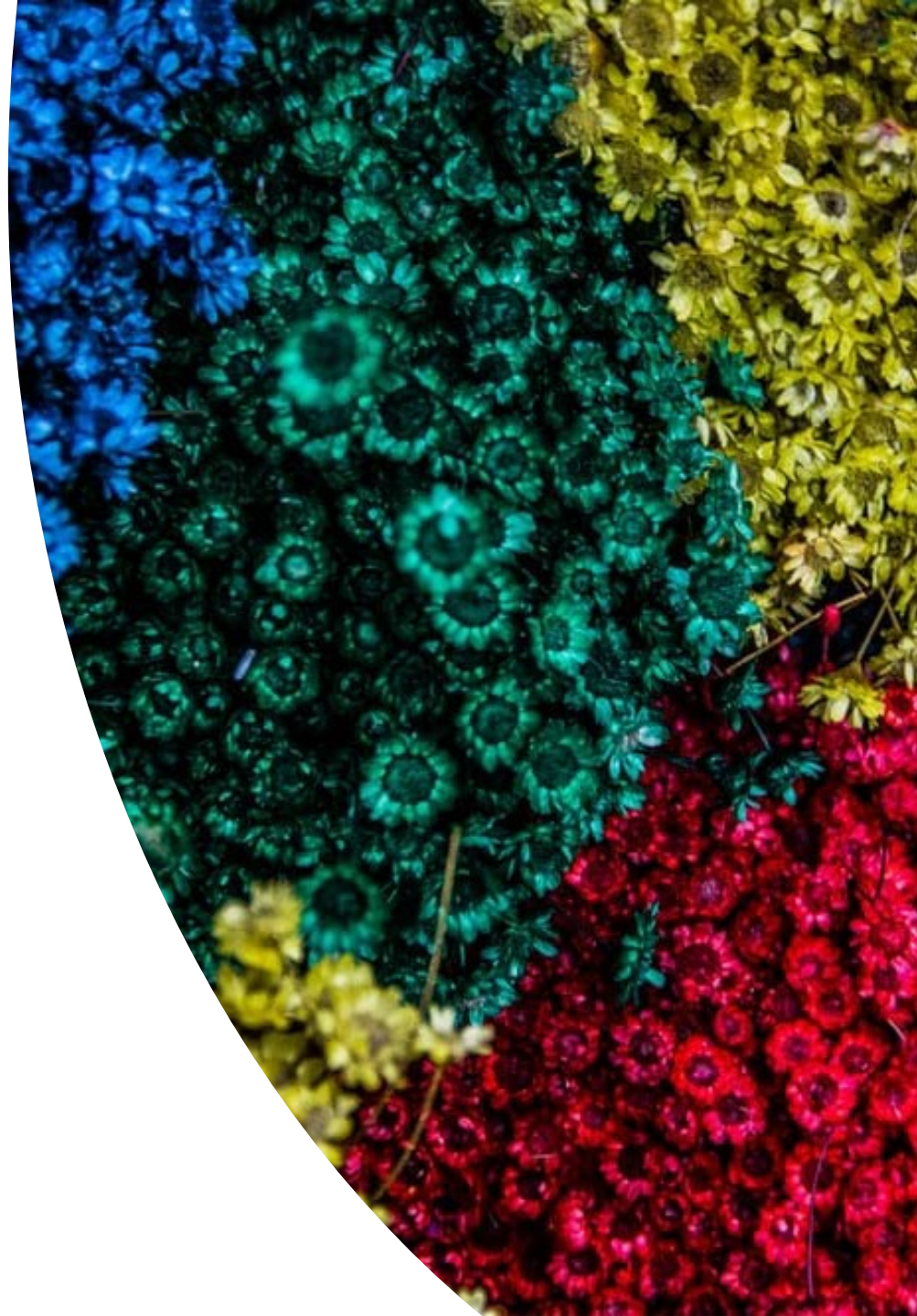
# Choosing k using the Average Silhouette Width

- Calculate the ASW for different values of k and choose the k with the maximum ASW (also called silhouette coefficient).



# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation**
- Outliers and Scaling Issues



# Measures for Cluster Evaluation

Numerical measures that are applied to judge various aspects of cluster quality, are classified into the following three types.

- **Internal Index (unsupervised evaluation):** Used to measure the goodness of a clustering structure without respect to external information.
  - E.g.: Sum of Squared Error (SSE), Silhouette coefficient, Correlation between proximity and incidence matrix
- **External Index (supervised evaluation):** Used to measure the extent to which cluster labels match externally supplied group labels.
  - E.g., Entropy, Purity, Rand index, ...
- **Relative Index:** Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

# External Measures of Cluster Validity: Entropy and Purity

**Table 5.9.** K-means Clustering Results for LA Document Data Set

| Cluster | Entertainment | Financial | Foreign | Metro | National | Sports | Entropy | Purity |
|---------|---------------|-----------|---------|-------|----------|--------|---------|--------|
| 1       | 3             | 5         | 40      | 506   | 96       | 27     | 1.2270  | 0.7474 |
| 2       | 4             | 7         | 280     | 29    | 39       | 2      | 1.1472  | 0.7756 |
| 3       | 1             | 1         | 1       | 7     | 4        | 671    | 0.1813  | 0.9796 |
| 4       | 10            | 162       | 3       | 119   | 73       | 2      | 1.7487  | 0.4390 |
| 5       | 331           | 22        | 5       | 70    | 13       | 23     | 1.3976  | 0.7134 |
| 6       | 5             | 358       | 12      | 212   | 48       | 13     | 1.5523  | 0.5525 |
| Total   | 354           | 555       | 341     | 943   | 273      | 738    | 1.1450  | 0.7203 |

**entropy** For each cluster, the class distribution of the data is calculated first, i.e., for cluster  $j$  we compute  $p_{ij}$ , the ‘probability’ that a member of cluster  $j$  belongs to class  $i$  as follows:  $p_{ij} = m_{ij}/m_j$ , where  $m_j$  is the number of values in cluster  $j$  and  $m_{ij}$  is the number of values of class  $i$  in cluster  $j$ . Then using this class distribution, the entropy of each cluster  $j$  is calculated using the standard formula  $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$ , where the  $L$  is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e.,  $e = \sum_{i=1}^K \frac{m_i}{m} e_j$ , where  $m_j$  is the size of cluster  $j$ ,  $K$  is the number of clusters, and  $m$  is the total number of data points.

**purity** Using the terminology derived for entropy, the purity of cluster  $j$ , is given by  $purity_j = \max p_{ij}$  and the overall purity of a clustering by  $purity = \sum_{i=1}^K \frac{m_i}{m} purity_j$ .

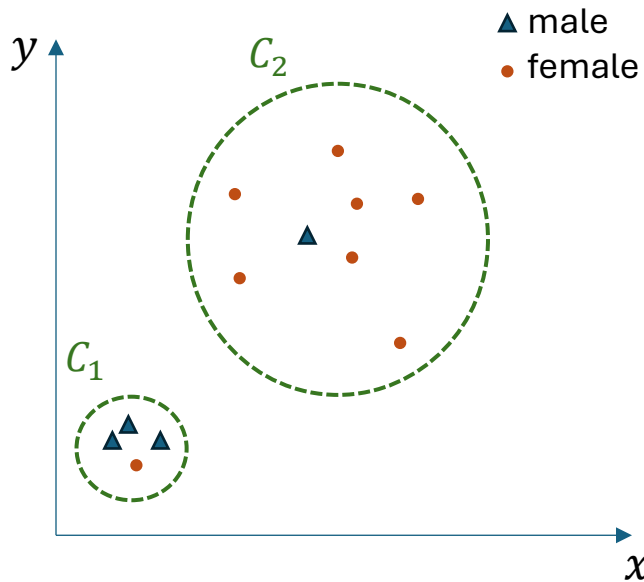
Other measures: Rand index/ARI, variation of information (VI) index

# Supervised Index: Purity

- Purity is a measure of the extent to which clusters contain a single class.
  1. For each cluster, count the number of data points from the majority class.
  2. Sum the count over all clusters and divide by the total number of data points.

- Example

$$purity = \frac{\# \text{ majority}}{\# \text{ total}}$$



|              | # male | # female | # total | # majority | Purity |
|--------------|--------|----------|---------|------------|--------|
| Cluster 1    |        |          |         |            |        |
| Cluster 2    |        |          |         |            |        |
| <b>Total</b> |        |          |         |            |        |

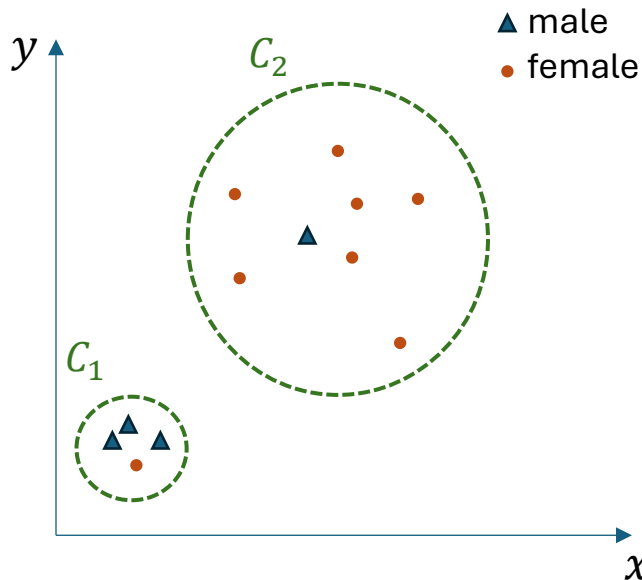


# Supervised Index: Purity

- Purity is a measure of the extent to which clusters contain a single class.
  1. For each cluster, count the number of data points from the majority class.
  2. Sum the count over all clusters and divide by the total number of data points.

- Example

$$purity = \frac{\# \text{ majority}}{\# \text{ total}}$$



|              | # male | # female | # total | # majority | Purity                   |
|--------------|--------|----------|---------|------------|--------------------------|
| Cluster 1    | 3      | 1        | 4       | 3          | $\frac{3}{4}$<br>= 75%   |
| Cluster 2    | 1      | 7        | 8       | 7          | $\frac{7}{8}$<br>= 87.5% |
| <b>Total</b> | 4      | 8        | 12      | 10         | $\frac{10}{12}$<br>= 83% |

- Note: Purity automatically increases when the number of clusters increases.

# Supervised Index: Rand Index

- The Rand index computes the similarity between two partitions in the range [0,1]. The partitions are the clusters and the ground truth classifications using pairwise comparisons.

$$RI = \frac{\# \text{ of concordant pairs} + \# \text{ of discordant pairs}}{\text{total \# of pairs}} = \frac{c+d}{\binom{n}{2}}$$

Concordant = objects are in both in the same cluster  
Discordant = objects are in a different cluster

- Example:  $O = \{A, B, C, D, E\}$

|           | Ground Truth | Clustering |
|-----------|--------------|------------|
| Cluster 1 | {A, B, C}    | {C, E}     |
| Cluster 2 | {D, E}       | {A, B, D}  |

$$RI = ?$$

Pairs:

$$c = ?$$

$$d = ?$$

**Adjusted Rand:** corrected for agreement by chance.

# Supervised Index: Rand Index

- The Rand index computes the similarity between two partitions in the range [0,1]. The partitions are the clusters and the ground truth classifications using pairwise comparisons.

$$RI = \frac{\# \text{ of concordant pairs} + \# \text{ of discordant pairs}}{\text{total \# of pairs}} = \frac{c+d}{\binom{n}{2}}$$

Concordant = objects are in both in the same cluster  
Discordant = objects are in a different cluster

- Example:  $O = \{A, B, C, D, E\}$

|           | Ground Truth | Clusterin g |
|-----------|--------------|-------------|
| Cluster 1 | {A, B, C}    | {C, E}      |
| Cluster 2 | {D, E}       | {A, B, D}   |

$$RI = \frac{3 + 1}{\binom{5}{2}} = \frac{4}{10} = 0.4$$

Pairs:

$(A, B)$  *c*  
 $(A, C)$        $(B, C)$   
 $(A, D)$        $(B, D)$        $(C, D)$  *d*  
 $(A, E)$  *d*       $(B, E)$  *d*       $(C, E)$        $(D, E)$

$c = 1$   
 $d = 3$

**Adjusted Rand:** corrected for agreement by chance.



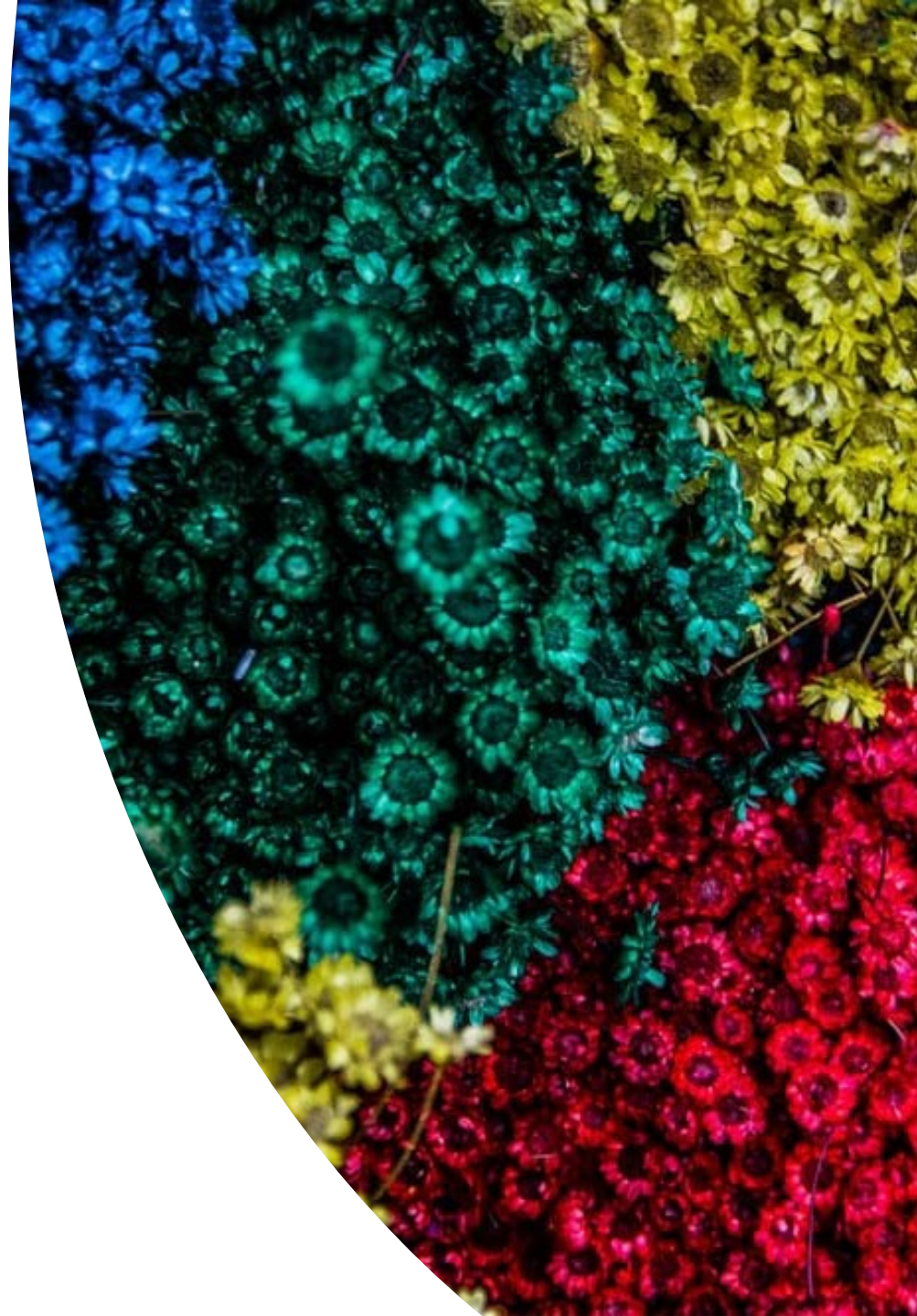
# Final Comment on Cluster Evaluation

“The validation of clustering structures is the most **difficult and frustrating** part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

*Jain and Dubes , Algorithms for Clustering Data, 1988*

# Topics

- Introduction
- Types of Clustering
- Types of Clusters
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Evaluation
  - Unsupervised Evaluation
  - Supervised Evaluation
- **Outliers and Scaling Issues**



# Outliers and Scaling Issues

- Clustering is based on **similarities/density** and therefore:
  - The features in your data need to be **scaled** to similar ranges.
    - For distance calculation: the feature with the largest range will dominate the distance.
    - For densities: a large range means that the densities will be artificially low.
    - **Note** that scatter plots scale the x and y-axis so the distances/densities you see is not what the algorithm calculates!
  - **Outliers** affect complete clustering algorithms. Outlier points may use their own cluster. You need to remove outliers or increase the number of clusters.
  - **Euclidean distance** may not always be the right way to measure similarity.



# Conclusion

- Clustering is an important method **to organize large data sets** into a small number of clusters. Cluster labels can be used as features in other data mining algorithms.
- **Meaning** is given to clusters by an expert by looking at the characteristics of the cluster (cluster profile). E.g., customers who live in urban areas and spend lots of money for Starbucks coffee.
- Clustering is based on **similarities/density** and therefore:
  - the features in your data needs to be **scaled** to similar ranges!
  - **Outliers** affect complete clustering algorithms.
  - **Euclidean distance** may not always be the right way to measure similarity.
- Deciding on the **number of clusters** and cluster evaluation is tricky!
- Data may not have a **clustering tendency**, but most algorithms will still return a clustering with the specified number of clusters.