We acknowledge the Australian Aboriginal and Torres Strait Islander peoples as the traditional owners of the lands and waters where we live and work.

MA5810 - Introduction to Data Mining

Week 4: Unsupervised Learning – Clustering

Dr Max Cao

Foundations for Data Science

- Week 1: Introduction
- Week 2: Supervised Learning – Bayes Classifiers
- Week 3: Supervised Learning - Logistic Regression & KNN
- Week 4: Unsupervised Learning – Clustering
- Week 5: Unsupervised Learning – Outlier Detection and PCA
- Week 6: Notions of Basket Analysis and Recommender Systems

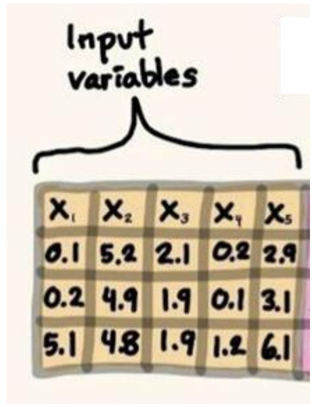
Unsupervised Learning

What: A type of machine learning that learns from data without human supervision

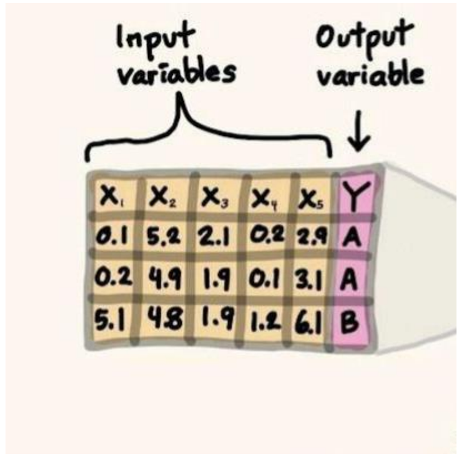
Goal: The goal is to model the underlying structure or distribution in the data in order to learn more about the data.

Main Types:

- **Clustering**: where you want to discover the inherent groupings in the data (e.g. grouping customers by purchasing behaviour)
- **Association**: where you want to discover relationships between data points in your dataset (e.g. people that buy X also tend to buy Y)
- **Dimensionality Reduction**: where you want to reduce the number of features in your dataset (e.g. PCA)


Input variables

X_1	X_2	X_3	X_4	X_5
0.1	5.2	2.1	0.2	2.9
0.2	4.9	1.9	0.1	3.1
5.1	4.8	1.9	1.2	6.1


Input variables
Output variable

X_1	X_2	X_3	X_4	X_5	Y
0.1	5.2	2.1	0.2	2.9	A
0.2	4.9	1.9	0.1	3.1	A
5.1	4.8	1.9	1.2	6.1	B

By: Chanin Nantasenamat

Unsupervised Learning- Clustering

- Partitional clustering
  - The k-means algorithm
- Clustering evaluation and model selection
  - The Silhouette Width Criterion (SWC)
- Hierarchical clustering
- Density-based clustering
  - Density-based spatial clustering of applications (DBSCAN)

Clustering

Definition: Clustering is an unsupervised data mining technique that groups similar observations together based on their intrinsic characteristics.

Objective: Discover hidden patterns or structures within unlabeled datasets. Observations should be:

- Similar to observations within the same cluster
- Different to observations in different clusters

Techniques:

- **K-means**: Divides observations into k clusters by minimizing the distance between observations and cluster centroids.
- **Hierarchical**: Builds a hierarchy of clusters by merging or splitting them based on their similarity.
- **Density-based (DBSCAN):** Identifies dense regions of observations and groups them as clusters.

Clustering

Applications:

- **Customer Segmentation**: Identify distinct groups of customers based on their purchasing behavior or demographic attributes.
- **Image Segmentation**: Group pixels in an image with similar color or texture characteristics.
- **Anomaly Detection**: Identify outliers or unusual patterns that deviate from normal behavior cases examples

Advantages:

- **Pattern Discovery**: Uncover natural groupings or structures in the data.
- **Exploratory Data Analysis**: Gain insights into the underlying distribution and relationships in the dataset.
- **Scalability**: Clustering algorithms can handle large datasets effciently.

Clustering – Real World Example

Business Understanding:

- **Objective:** Divide customers into distinct groups based on their purchasing behavior.
- **Dataset:** Customer transaction data containing information such as purchase history, demographics, and browsing patterns.
- **Technique:** K-means Clustering

Process:

Preprocess the data: Normalize or scale the features to ensure comparability.
Determine the number of clusters (k): Use techniques like the elbow method or silhouette analysis.
Run the K-means algorithm: Randomly initialize k cluster centroids and iteratively assign instances to the nearest centroid, updating the centroids' positions.
Evaluate the results: Analyze the within-cluster sum of squares (WCSS) or silhouette score to assess cluster quality.
Interpret the clusters: Examine the characteristics and behaviors of customers within each cluster.

Clustering – Real World Example

Example Results:

- **Cluster 1**: Young, tech-savvy customers who frequently make online purchases.
- **Cluster 2**: Budget-conscious customers who prefer in-store shopping and seek discounts.
- **Cluster 3**: High-spending luxury shoppers who purchase premium products.
- **Cluster 4**: Occasional buyers who make infrequent purchases.

Insights and Actions:

Tailor marketing strategies: Develop targeted campaigns for each customer segment.
Personalize recommendations: Recommend products based on each segment’s preferences.
Optimize inventory: Stock inventory based on the demand patterns of different clusters.

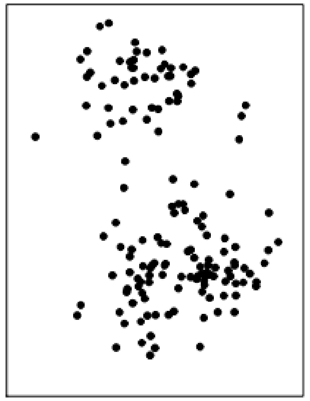
Partitional clustering

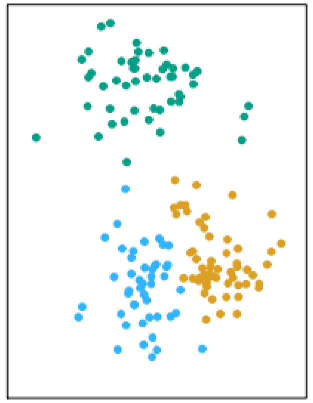
- Clustering is a key unsupervised task to group observations into clusters
- We are interested in identifying groups where observations within a cluster are more similar to each other than to those in other clusters.
- In partitional clustering, we essentially want to categorise the data observations into disjoint subsets.
- We have n observations

$$P = \{C_1, C_2, , , , , C_k\}$$

- $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \cdots \cup \mathbf{C}_k = \mathbf{X}$
- $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset \; (\forall i \neq j)$
- $\mathbf{C}_i \neq \emptyset \; (\forall i).$

$|C_i|$ is the number of observations in the cluster





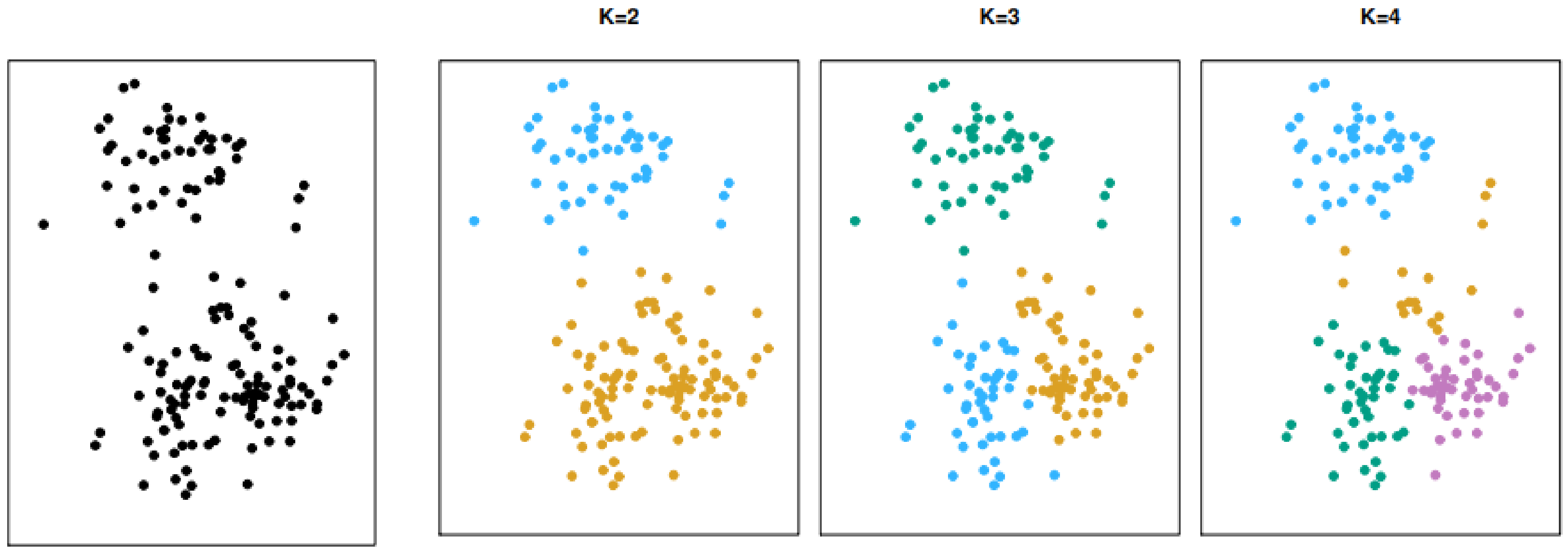
What is good clustering?

High -intra class similarity: Cohesive within clusters

Low inter-class similarity: Distinctive between clusters

The k-means algorithm

- **K-means Clustering** is the most well known yet simple method for partitioning data into k clusters.
- We have K distinct, non-overlapping groups.
- We need to specify the number of clusters for the algorithm (k).



The k-means algorithm

- Textbook: “The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible.”
- We want the observations within cluster to be as similar to each other as possible.
- How to measure similarity? Euclidean distances.

The k-means algorithm

- How does the k-means algorithm work?
- Adjusts k cluster prototypes (centroid), represent each cluster, as well as possible.
- How?
- By minimizing the sum of squared Euclidean distances between cluster points and their prototype.

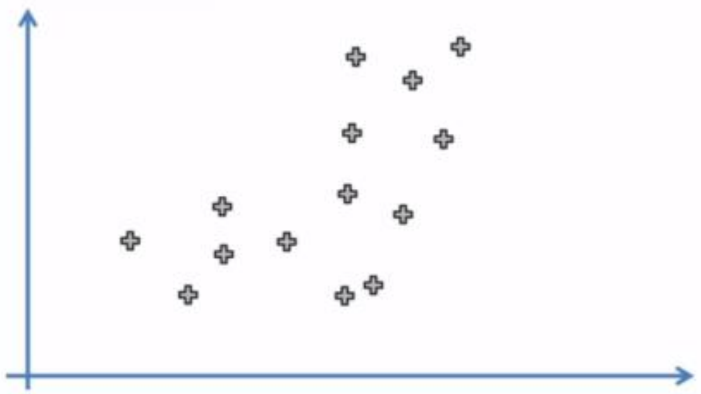
$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, \overline{x_i})^2 = \sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j - \overline{x_i})^2$$

- So we select k points, $\overline{x_1}, \dots, \overline{x_k}$ in a way that the SSE is minimized.

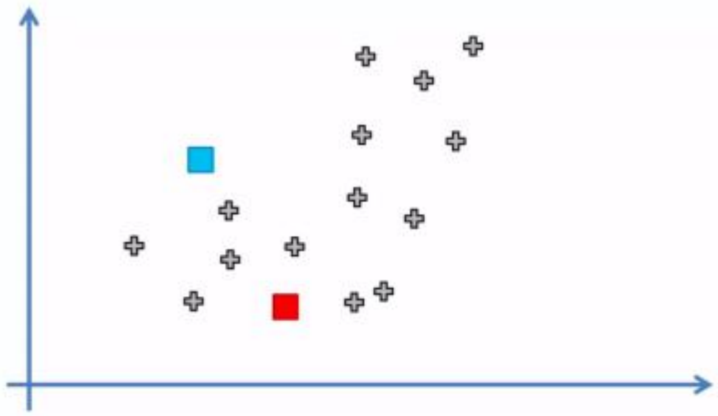
The k-means algorithm

1. Choose k
2. Select k prototypes randomly.
3. Assign each of the n observations in the dataset to its closest prototype (closest is defined using the Euclidean distances)
4. Recalculate the prototypes
5. Repeat to steps 3 and 4 (Until nothing changes from the previous iteration or until a maximum number of iterations is reached)

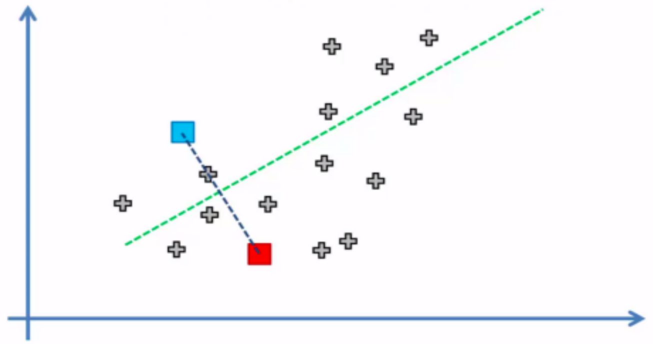
STEP 1: Choose the number K of clusters: K = 2

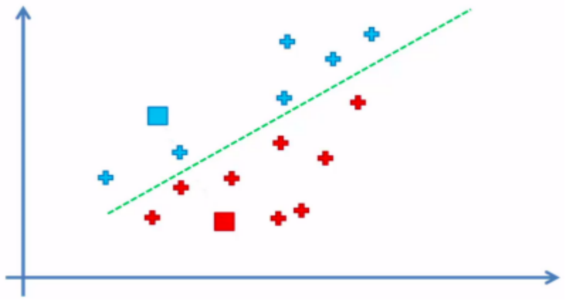


STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

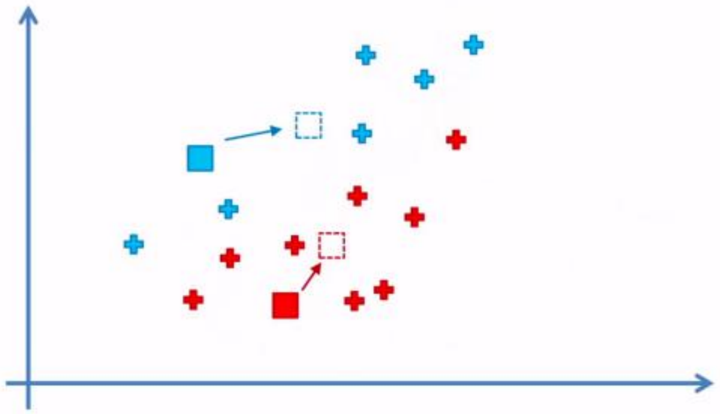


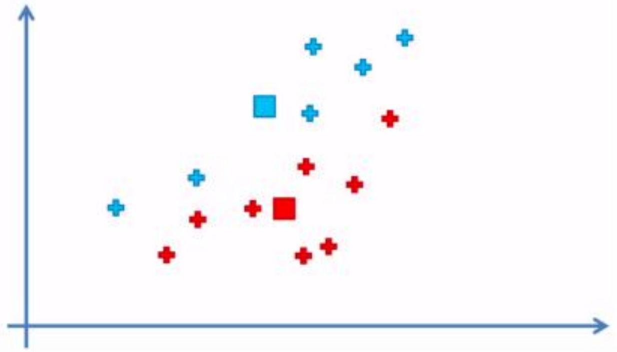
STEP 3: Assign each data point to the closest centroid ➡ That forms K clusters

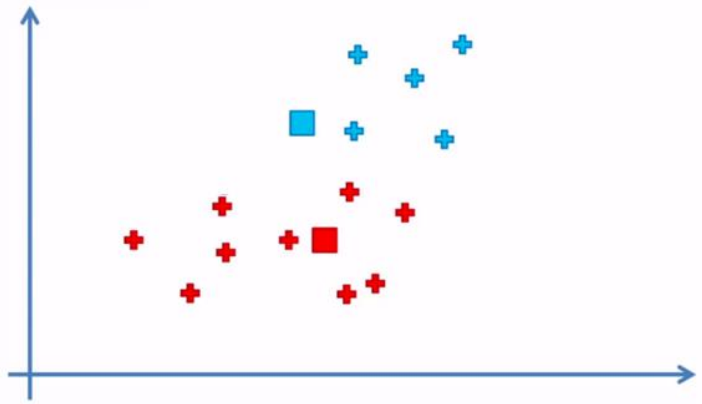




STEP 4: Compute and place the new centroid of each cluster







Repeating K-Means Clustering

K-means clustering is an iterative algorithm that partitions data into K clusters, where each data point belongs to the cluster with the nearest mean (centroid).

The algorithm proceeds through several iterations until convergence, where the cluster assignments stabilize, and the centroids stop changing significantly.

However, even after convergence, there are reasons why you might need to repeat K-means clustering.

It's important to note that in K-means the results obtained can vary depending on the dataset and initialization. Repeating K-means clustering with different settings or multiple runs allows for a more comprehensive exploration of the solution space and helps increase the chances of finding a better clustering solution.

Repeating K-Means Clustering

Initialization Sensitivity: K-means clustering is sensitive to the initial placement of the centroids. Different initializations can lead to different clustering results. To overcome this, the algorithm often performs multiple runs with different initializations and chooses the clustering that minimizes the sum of squared distances between data points and their respective centroids (inertia).

Local Optima: K-means can converge to local optima, where the clustering might not be the best possible solution. By repeating the clustering process with different initializations or variations, you increase the chances of finding a better clustering arrangement.

Large Datasets: For large datasets, running K-means once might not be sufficient to explore the full solution space. Running K-means multiple times with random initialization can help ensure a better representation of different possible clusters.

Changing Data: If your data is non-stationary and subject to change over time, it may be necessary to re-run K-means periodically to adapt the clusters to the updated patterns in the data.

Model Evaluation: When comparing di"erent clustering algorithms or assessing the performance of K-means under different parameters, multiple runs can provide more robust evaluation metrics, such as average silhouette score, to select the best model.

Ensemble Clustering: In ensemble clustering, multiple K-means runs are combined to create a more reliable clustering result. This technique can help reduce the impact of random initialization and improve the overall clustering quality.

Scaling

Scaling (also known as normalization or standardization) is a common preprocessing step in K-means clustering, and many other clustering algorithms as well. Scaling is done to ensure that all features have equal importance during the clustering process. There are several reasons why scaling is important in K-means clustering:

Different Scales: K-means clustering uses the distance between data points and cluster centroids to assign points to clusters. If the features have different scales or units, those with larger scales can dominate the clustering process, making the clusters sensitive to the choice of units. Scaling standardizes the features to a common scale, preventing this dominance issue.

Equal Variance: K-means assumes that the variance of each feature is equal. If features have different variances, the algorithm will be biased toward features with larger variances. Scaling helps achieve equal variance and ensures that all features contribute equally to the clustering.

Convergence Speed: Scaling can improve the convergence speed of the K-means algorithm. When features have different scales, the clusters might converge slowly, as the algorithm will take more iterations to reach convergence. By scaling the features, the convergence speed can be accelerated.

Numerical Stability: Some distance metrics, like the Euclidean distance, can be numerically unstable when the data has widely varying scales. Scaling the data can help stabilize the computation of distances and centroids.

Interpretability: Scaling makes the interpretation of cluster centroids more straightforward. Since all features are on the same scale, the coordinates of the centroids represent the average values of the features in each cluster, making them more interpretable.

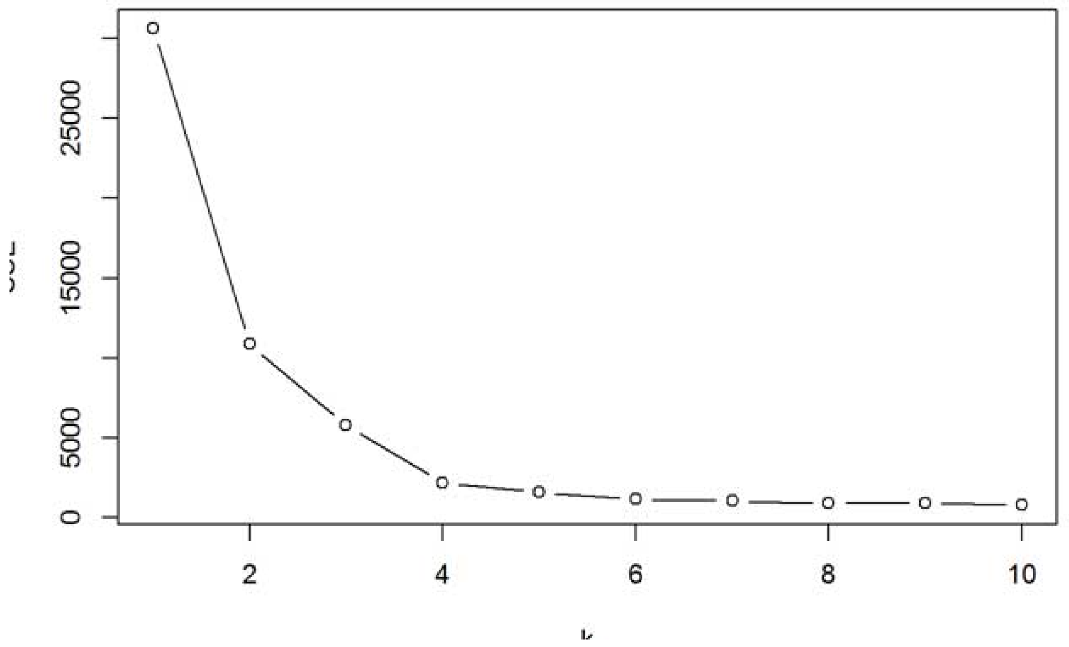
There are different scaling methods available, such as z-score normalization (subtracting the mean and dividing by the standard deviation) and Min-Max scaling (scaling the data to a specific range, often [0, 1]). The choice of scaling method depends on the specific characteristics of the data and the requirements of the clustering algorithm.

Clustering evaluation and model selection

- Problem: We don't have a reference or ground truth solution to assess the performance of the results.
- How to choose parameters? -> k=?
    - Run k-means several times for varied k and choose the value of k that provides the best (minimum) SSE?
    - The problem is SSE decreases when k increases, no matter what:
    - **Elbow Method**
    - **The Silhouette Width Criterion (SWC)**

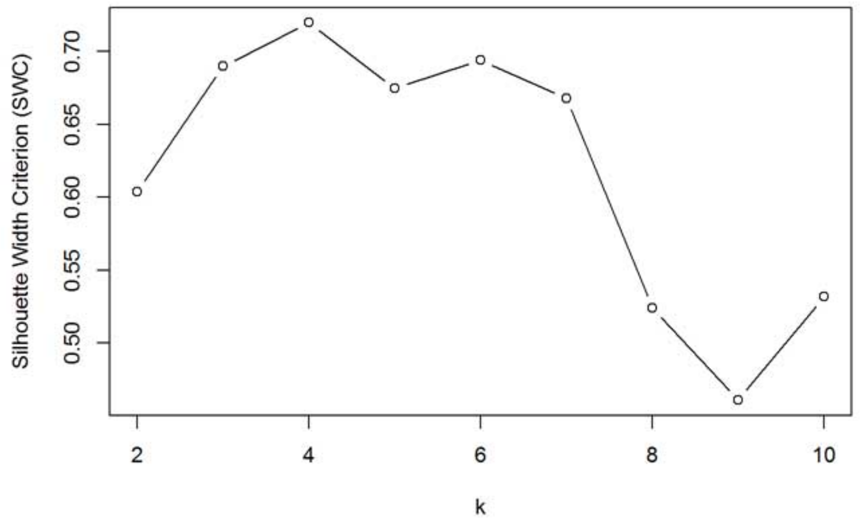
Elbow Method

Plotting the within-cluster sum of squares (inertia) against different values of K and looking for the "elbow" point on the graph, where the inertia decreases sharply. The idea is that the inertia will decrease as k increases, but at a certain point, the rate of decrease will slow down. The k value at the elbow is often considered to be a reasonable choice.



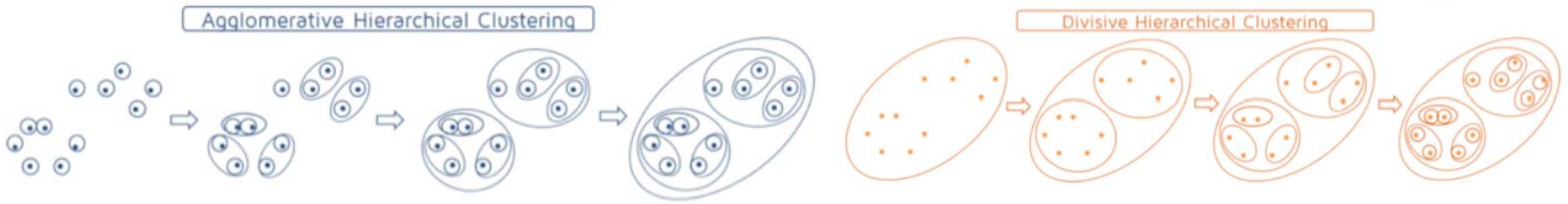
The Silhouette Width Criterion (SWC)

- SWC evaluates how well each point fits within its cluster compared to other clusters, by measuring the difference between the average distance of this observation to observations in another cluster (which should be large) and the average distance of this observation to observations in the same cluster (which should be low).
- A higher silhouette score indicates that the clusters are well-defined and appropriately separated.
- We can calculate the silhouette score for different values of k and choose the one with the highest score.



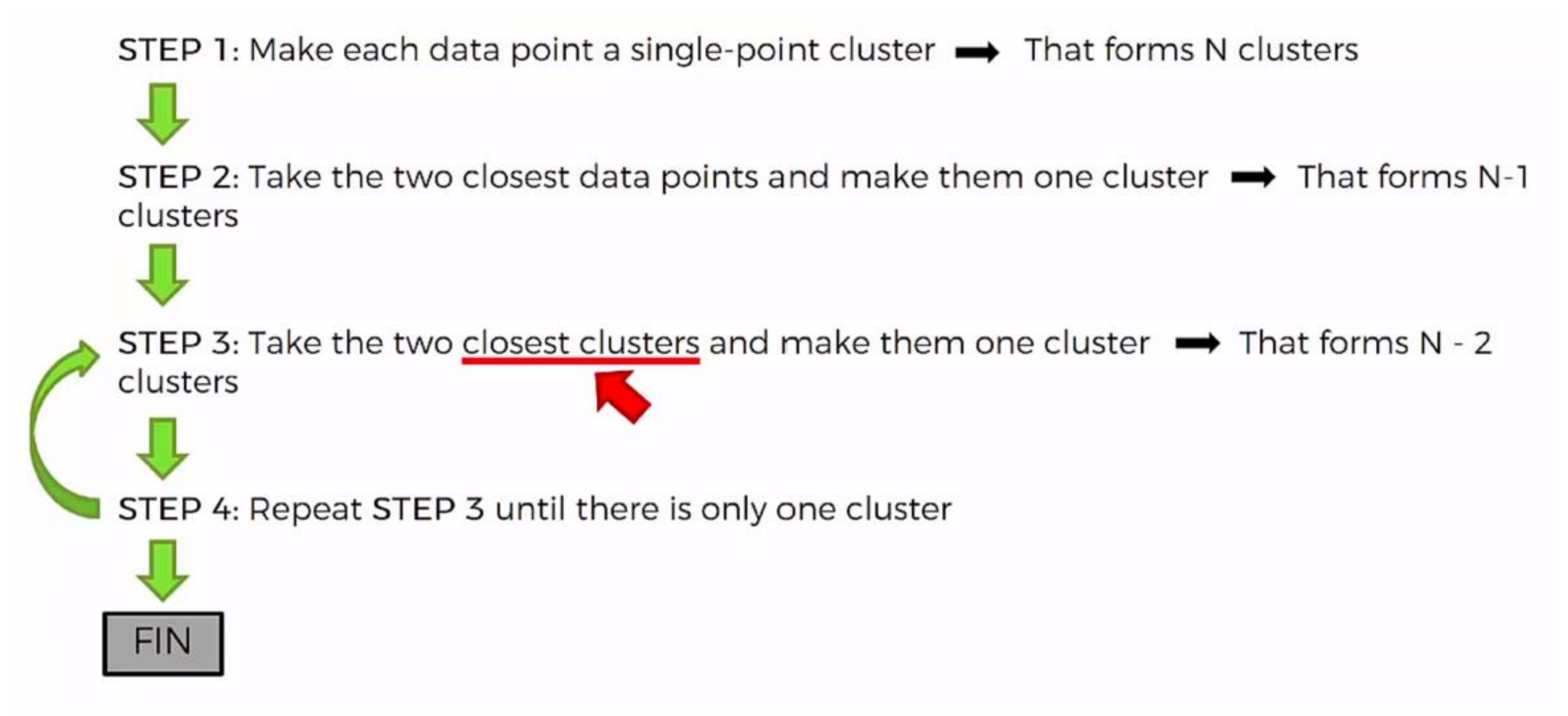
Hierarchical Clustering

- Hierarchical clustering produce a complete collection of clustering solutions with varied numbers of clusters, rather than a single solution
- We do not need to specify k
- **Divisive Hierarchical Clustering** (DIANA- Divisive Analysis) is also termed as a top-down clustering approach. In this technique, entire data or observation is assigned to a single cluster. The cluster is further split until there is one cluster for each data or observation.
- **Agglomerative Hierarchical Clustering** (AGNES- Agglomerative Nesting) is popularly known as a bottom-up approach, wherein each data or observation is treated as its cluster. A pair of clusters are combined until all clusters are merged into one big cluster that contains all the data.


Agglomerative Hierarchical Clustering
Divisive Hierarchical Clustering

Source:https://quantdare.com/hierarchical-clustering

Agglomerative Hierarchical Clustering



STEP 1: Make each data point a single-point cluster ➡ That forms N clusters

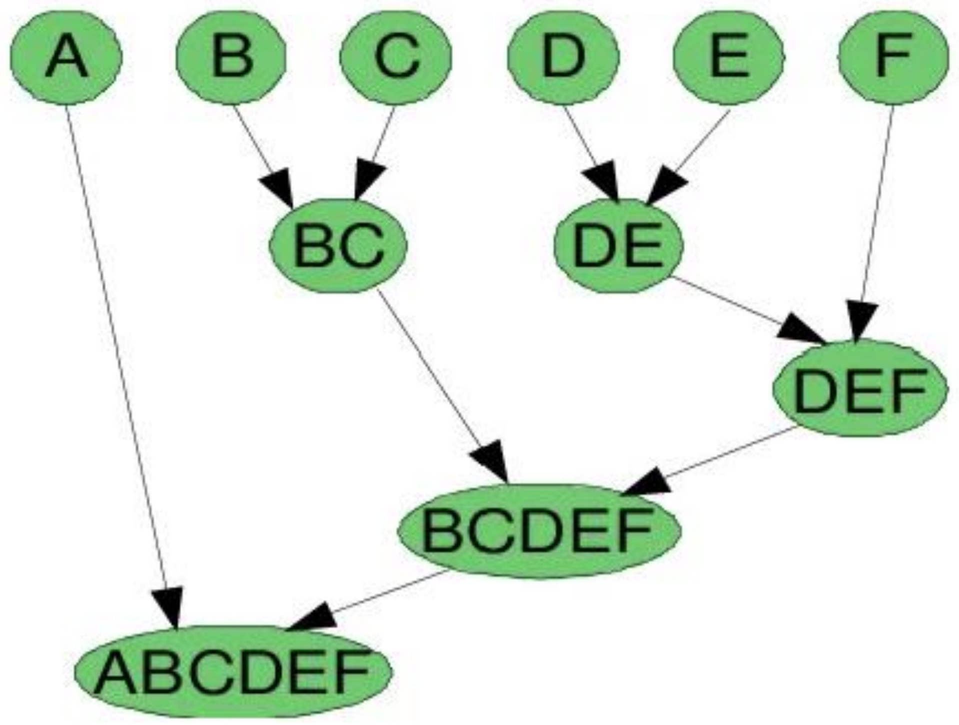
STEP 2: Take the two closest data points and make them one cluster ➡ That forms N-1 clusters

STEP 3: Take the two closest clusters and make them one cluster ➡ That forms N - 2 clusters

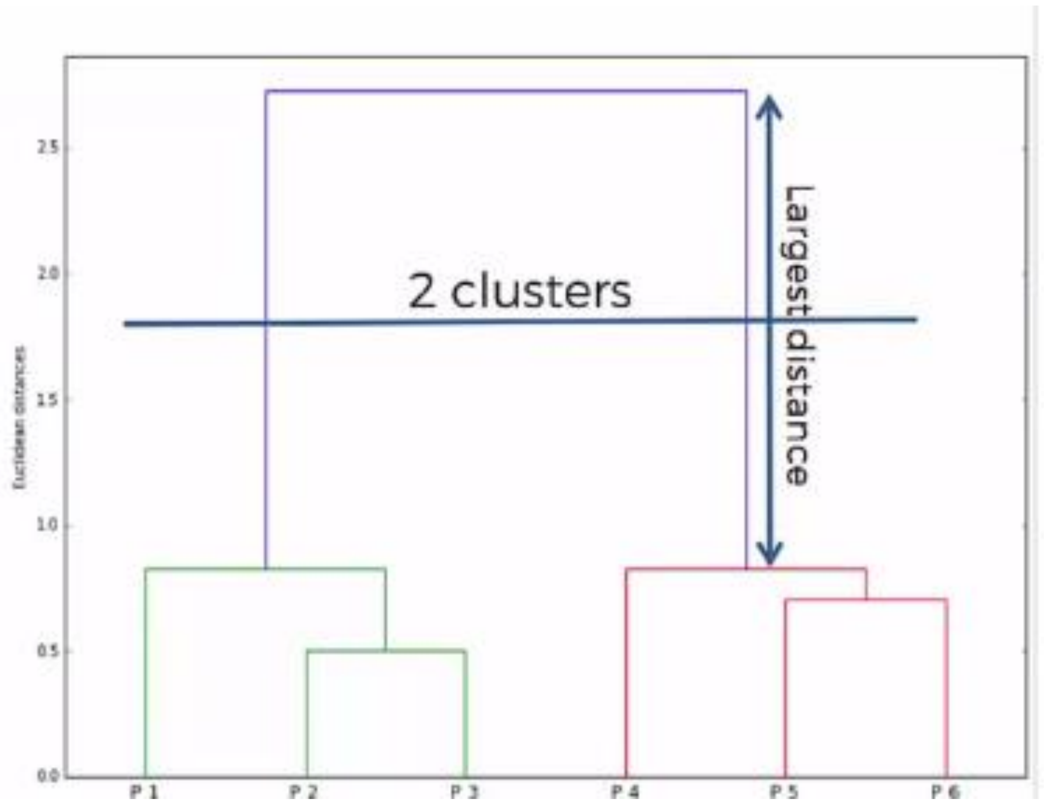
STEP 4: Repeat STEP 3 until there is only one cluster

FIN

Agglomerative Hierarchical Clustering



AHC

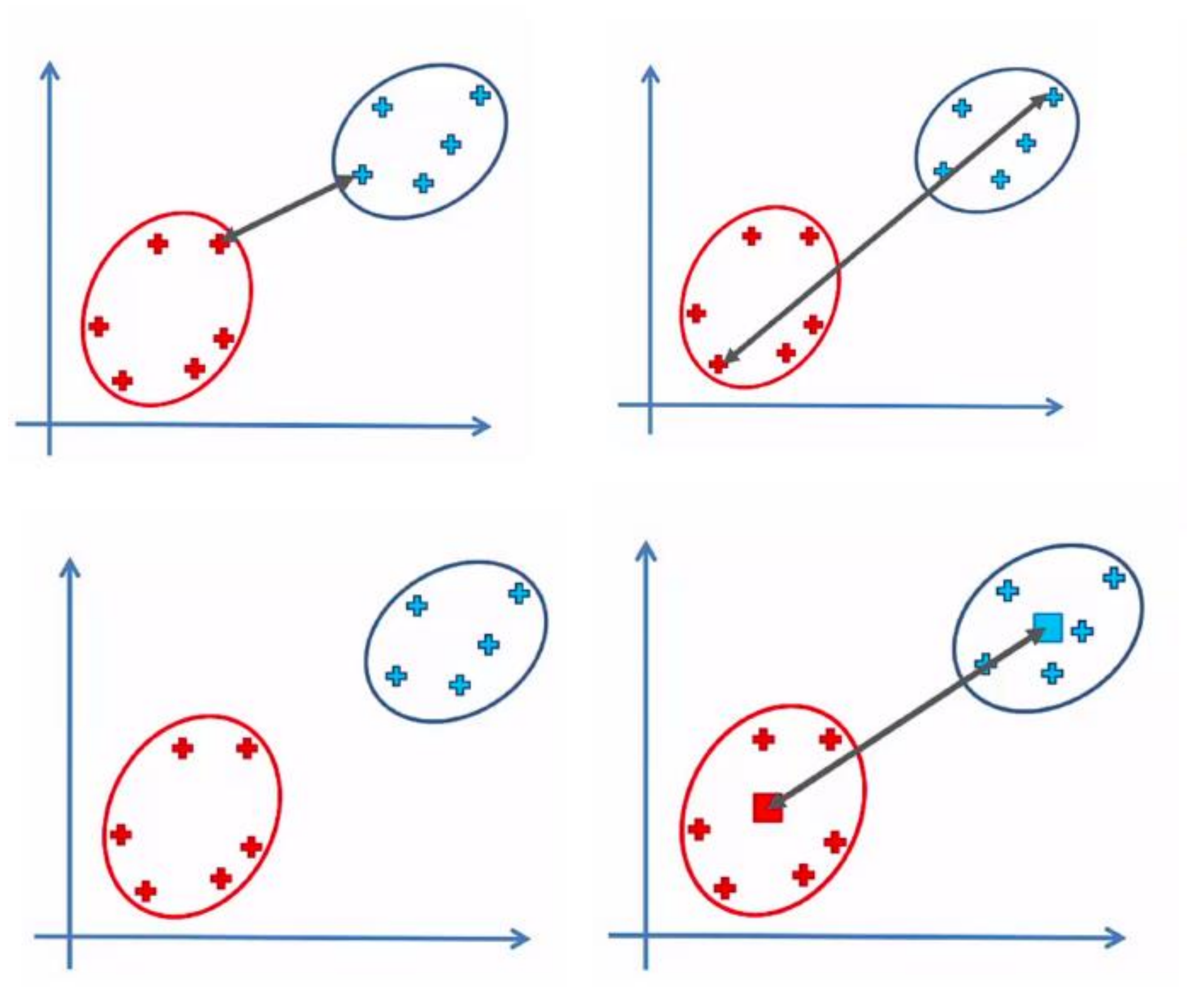


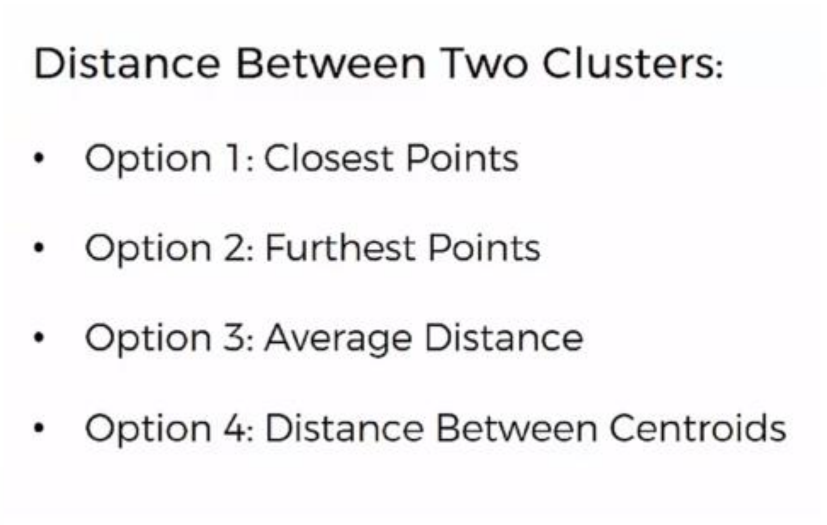
Dendrogram representation

Hierarchical Clustering

- How do we choose the pair of clusters to merge at each iteration? – Closest Clusters
- The similarity measure, such as Euclidean distance, can be used, but since each cluster contains multiple observations, we have a number of similarity measures.
- We need a **linkage criterion** that determines how to measure the similarity between clusters when there are multiple observations in each cluster.
- **Distance**: Any suitable distance metric, such as Euclidean distance, can be used.

Hierarchical Clustering





Distance Between Two Clusters:

- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance
- Option 4: Distance Between Centroids

Ward's algorithm
If two clusters are merged, the variance of the resulting cluster will not be smaller than the variance taking the two individual clusters separately

Hierarchical Clustering

Strength:

- No need to assume any particular number of clusters
- Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
- Traditional hierarchical algorithms use similarity or distance to merge or split one cluster at a time.
- They may correspond to meaningful taxonomies (example: animal kingdom)

Weakness:

- Hierarchical Clustering only makes one pass through the data. Therefore, early clustering decisions affect the rest of the clustering results.
- Hierarchical Clusters often have low stability. That is, adding or subtracting variables or adding or dropping observations can affect the groupings substantially.
- The determination of final clusters can be sensitive to outliers and their treatment.

Density-based clustering - DBSCAN

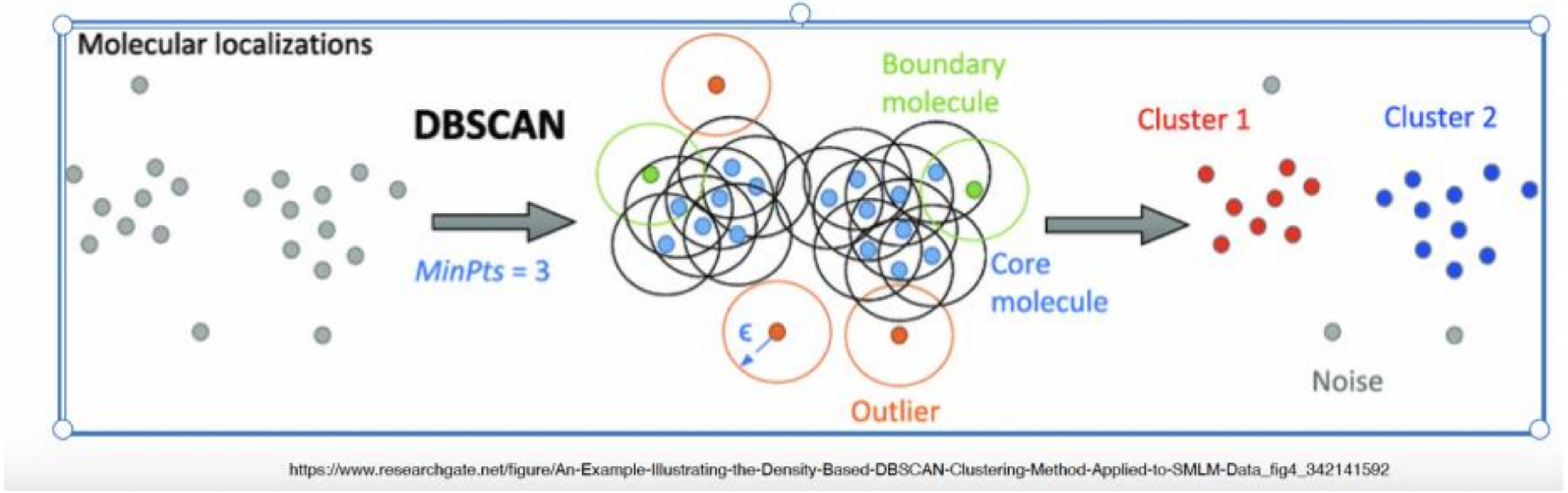
Density-based spatial clustering of applications with noise (DBSCAN) groups together points that are

- close to each other based on a distance measurement (usually Euclidean distance) and
- a minimum number of points

It marks as outliers the points that are in low-density regions.

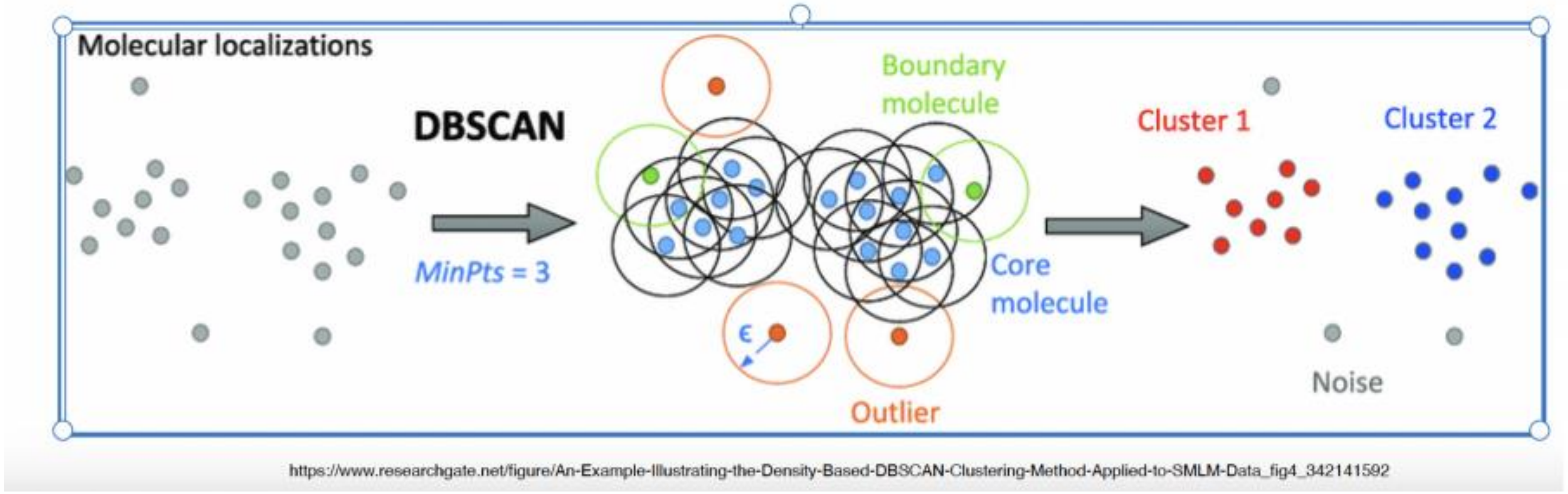
Parameters:

- **eps**: the minimum distance between two points. if the distance between two points is lower or equal to this value (eps), these points are considered neighbours
- **minPoints**: the minimum number of points to form a dense region.


Molecular localizations
DBSCAN
MinPts = 3
Boundary molecule
ϵ
Core molecule
Outlier
Cluster 1
Cluster 2
Noise
https://www.researchgate.net/figure/An-Example-Illustrating-the-Density-Based-DBSCAN-Clustering-Method-Applied-to-SMLM-Data_fig4_342141592

Density-based clustering - DBSCAN

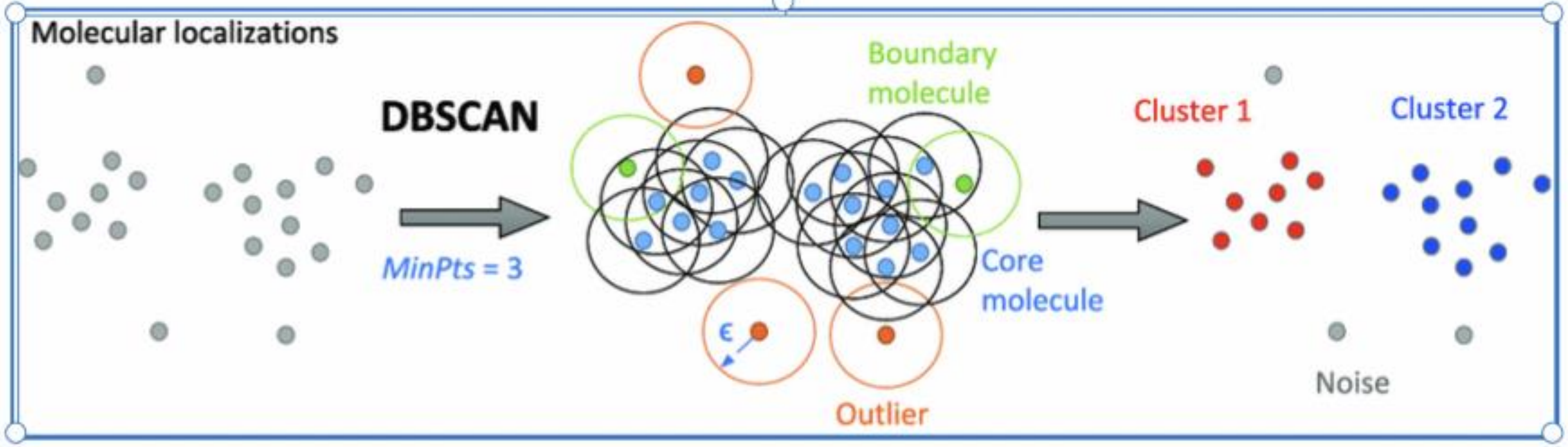
- An observation x_i is considered a core point if there are at least minPts (minimum points) observations within its ϵ-neighborhood.
- Core points within each other's eps-neighborhood are connected and form the **same cluster.**
- Clusters are maximal sets of connected core points.
- Border points are non-core observations within the eps-neighborhood of a core point and are part of the cluster.
- Remaining observations that are neither core nor border points are considered noise and treated as outliers, not included in any cluster.



https://www.researchgate.net/figure/An-Example-Illustrating-the-Density-Based-DBSCAN-Clustering-Method-Applied-to-SMLM-Data_fig4_342141592

Density-based clustering - DBSCAN

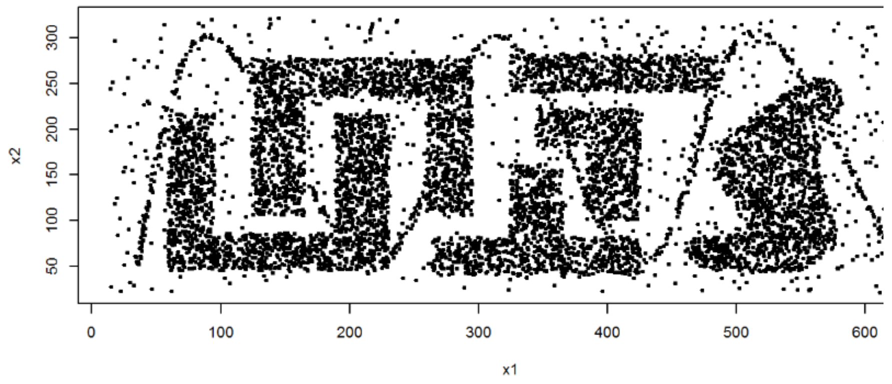
1. **Identify Core Points**: For each point in the dataset, count the number of points within its **eps** neighborhood. If the count meets or exceeds **MinPts**, mark the point as a **core point**.
2. **Form Clusters**: For each core point that is not already assigned to a cluster, create a new cluster. Recursively find all **density-connected points** (points within the **eps** radius of the core point) and add them to the cluster.
3. **Density Connectivity**: Two points, **a** and **b**, are **density-connected** if there exists a chain of points where each point is within the **eps** radius of the next, and at least one point in the chain is a core point. This chaining process ensures that all points in a cluster are connected through a series of dense regions.
4. **Label Noise Points**: After processing all points, any point that does not belong to a cluster is labeled as **noise**.

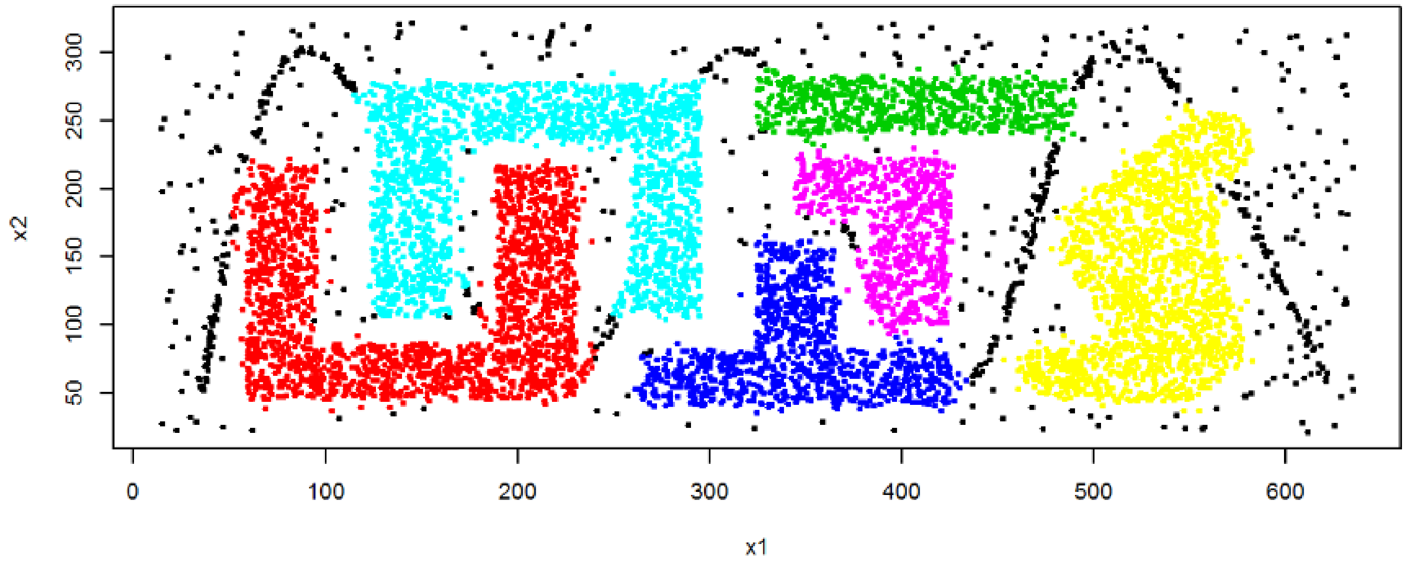


https://www.researchgate.net/figure/An-Example-Illustrating-the-Density-Based-DBSCAN-Clustering-Method-Applied-to-SMLM-Data_fig4_342141592

Density-based clustering - DBSCAN

- DBSCAN can identify shapes in a dataset.
- Unlike to K-means, DBSCAN does not require the user to specify the number of clusters
- DBSCAN can find any shape of clusters. The cluster doesn't have to be a circle.
- DBSCAN can identify outliers.





Enough chatter- Let’s get down to R!