Machine Learning

Unsupervised Learning

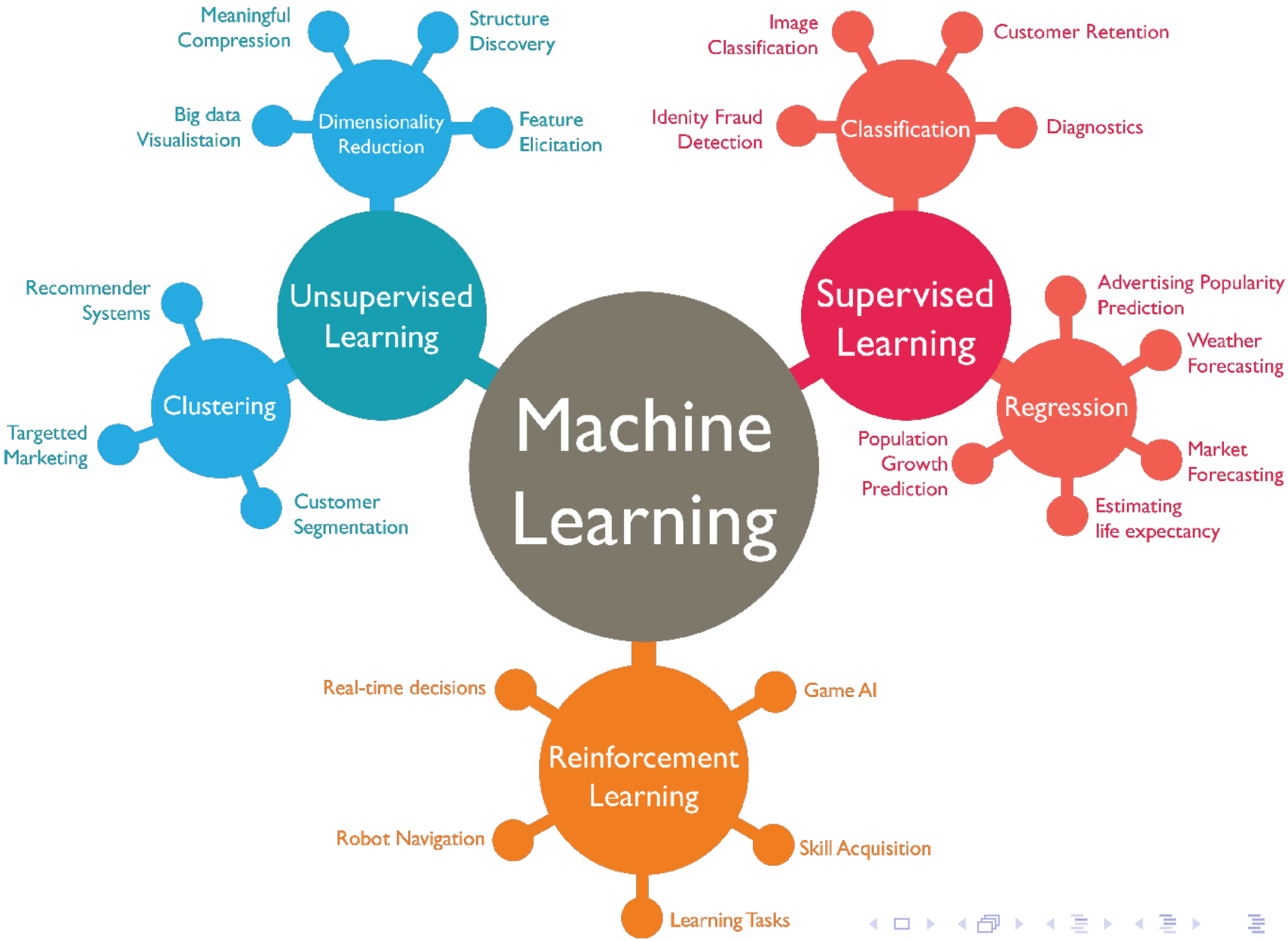
Topics of previous lectures

- ✓ Ingredients of Machine Learning
- ✓ Classification Basics, Basic Linear Classifier
- ✓ K-Nearest Neighbours and Naive Bayes Classifier
- ✓ Linear and Quadratic Discriminant Analysis
- ✓ Support Vector Machines (SVM)
- ✓ Decision Trees
- ✓ Ensemble Methods (Bagging, Weighted Voting, Stacking)
- ✓ Regression Methods
- ✓ Evaluation and Scoring of Classifiers
- ✓ Ensemble Methods (Boosting)

Topics of today's lecture

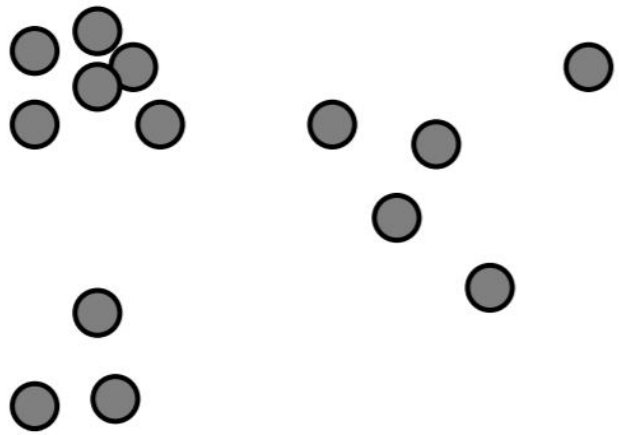
- Concepts of Cluster Analysis
- Hierarchical Clustering
- K-means Clustering
- K-medoids Clustering
- DBSCAN
- Dissimilarity Measure Selection

Machine Learning Map



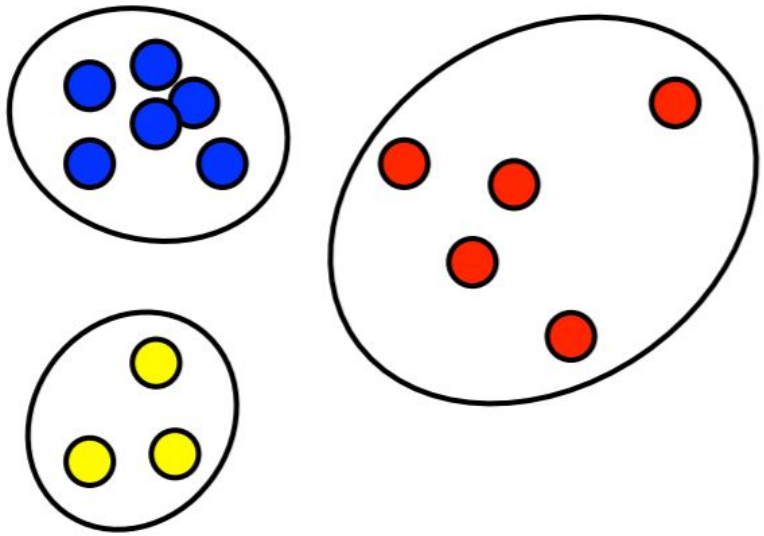
What is clustering?

- Given data objects



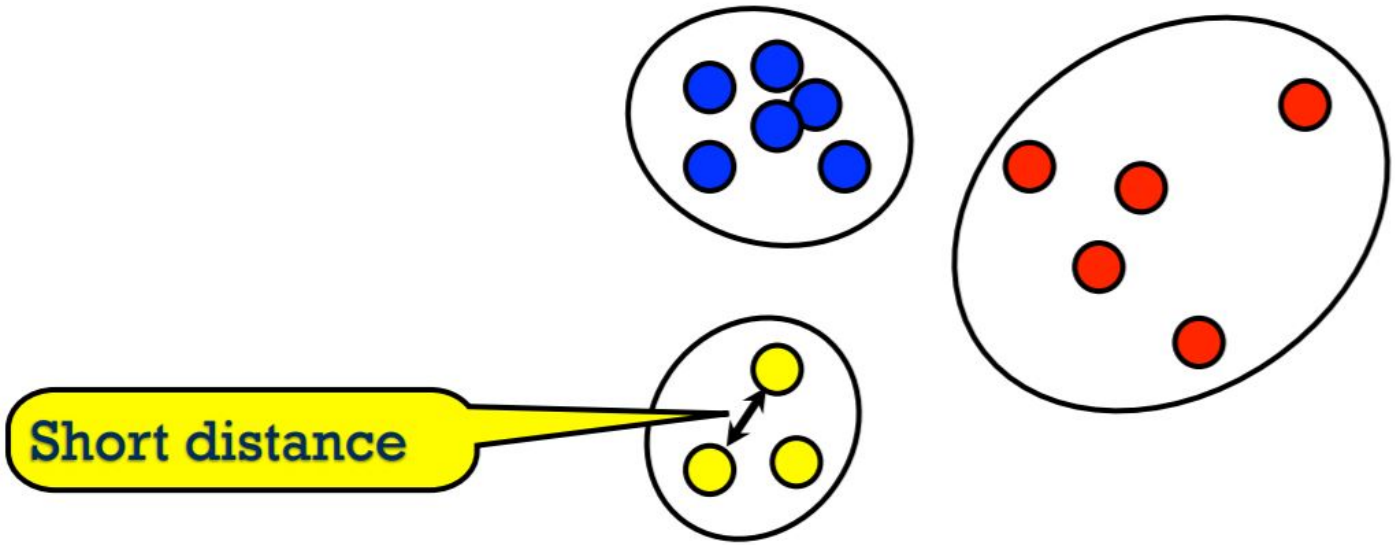
What is clustering?

- Given data objects
- Find a grouping (**clustering**) such that the objects are:



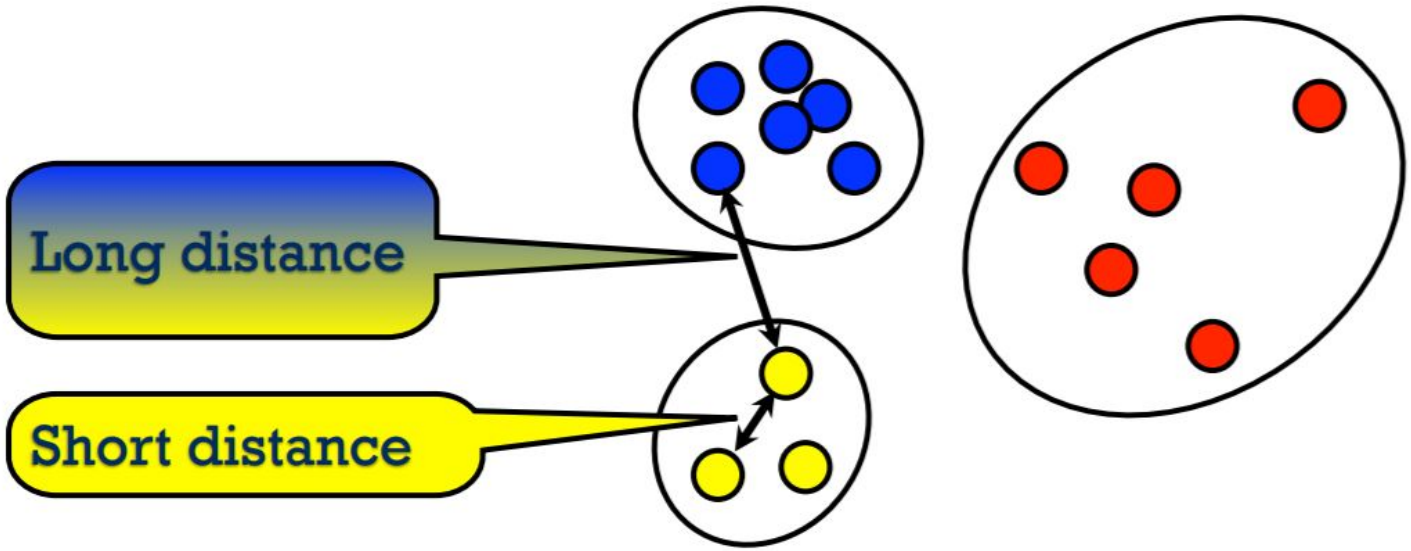
What is clustering?

- Given data objects
- Find a grouping (**clustering**) such that the objects are:
  - similar (related) to the objects in the same group



What is clustering?

- Given data objects
- Find a grouping (**clustering**) such that the objects are:
  - similar (related) to the objects in the same group
  - dissimilar (unrelated) from objects in other groups



Why to cluster data?

- Intuition building

Why to cluster data?

- Intuition building
- Hypothesis generation

Why to cluster data?

- Intuition building
- Hypothesis generation
- Discover structures and patterns in high-dimensional data

Why to cluster data?

- Intuition building
- Hypothesis generation
- Discover structures and patterns in high-dimensional data
- Summarizing / compressing large data

Clustering is subjective

- Suppose that we need to put you in some groups for projects

Clustering is subjective

- Suppose that we need to put you in some groups for projects
- How to define those groups?
  - Work experience
  - Age
  - Education
  - Preferences

Clustering is subjective

- Suppose that we need to put you in some groups for projects
- How to define those groups?
  - Work experience
  - Age
  - Education
  - Preferences
- Which way is correct? Depends on the goal!

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation
- Search result grouping

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation
- Search result grouping
- Social network analysis

Examples

- Google news uses the clustering algorithm to categories the news related to the same topic
- Segmentation of the people according to the items purchased in e-commerce applications
- Segmentation of the customers of the fashion company
- Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables
- Find stocks which have common price fluctuations
- Image segmentation
- Search result grouping
- Social network analysis
- Anomaly detection

Example: Supermarket data

Customer 1

C

Task

Cluster customers into segments by their typical shopping baskets

Customer 3

Customer 4

Example: Supermarket data

Customer 1

C

Task

Cluster customers into segments by their typical shopping baskets

Customer 3

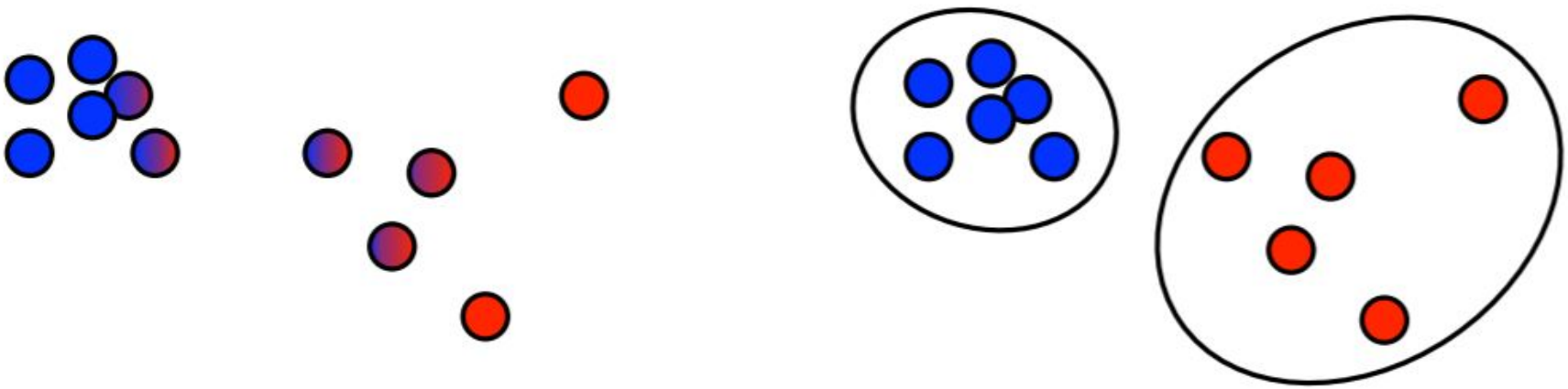
C

Motivation

Launch a discount campaign addressing a particular segment of customers

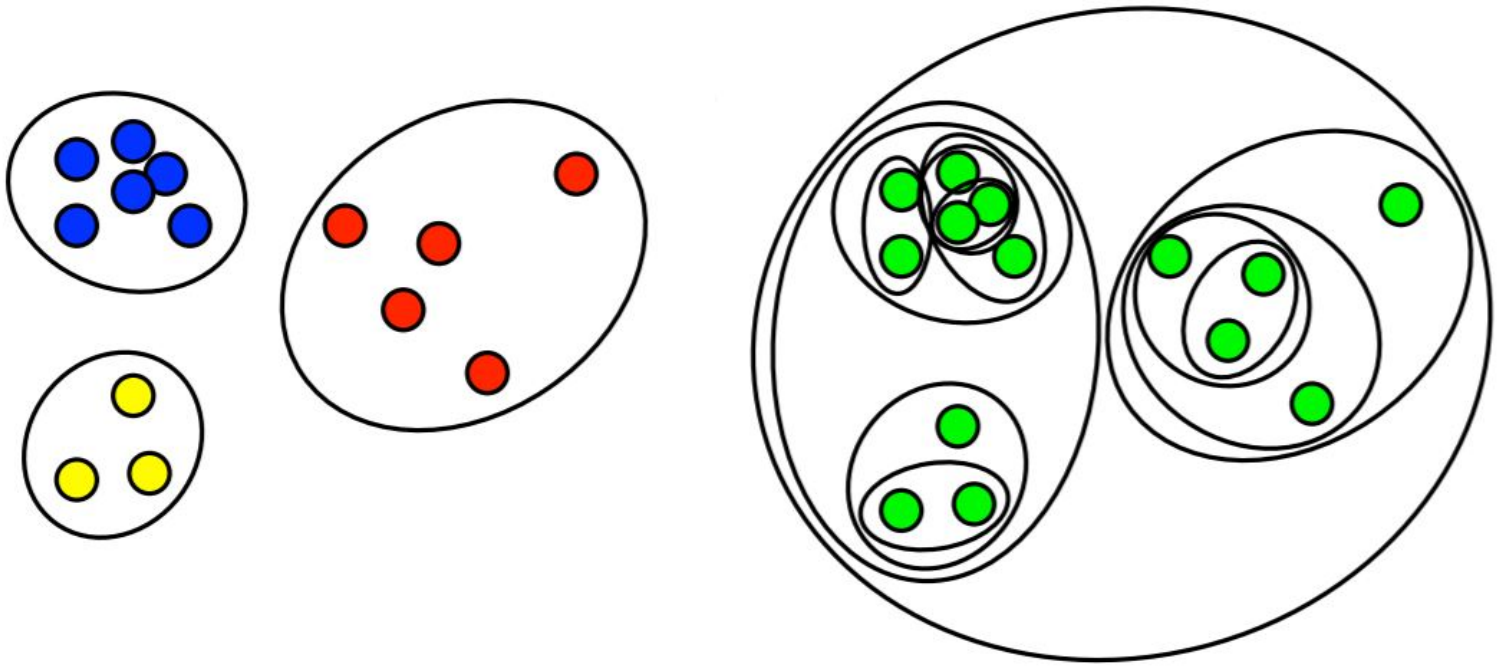
Clustering Types

- **Fuzzy (Soft) clustering**: Each object belongs to each cluster with some weight (the weight can be zero)
- **Non-fuzzy (Hard) clustering** each object belongs to exactly one cluster



Clustering Types

- **Partitional clustering** finds a fixed number of clusters
- **Hierarchical clustering** creates a series of clusterings contained in each other (nested clusters)



Hierarchical Clustering

- Hierarchical clustering is usually visualized as a dendrogram (tree)

Hierarchical Clustering

- Hierarchical clustering is usually visualized as a dendrogram (tree)
- Each subtree corresponds to a cluster

Hierarchical Clustering

- Hierarchical clustering is usually visualized as a dendrogram (tree)
- Each subtree corresponds to a cluster
- Height of branching shows distance between the objects

Hierarchical Clustering

- Hierarchical clustering is usually visualized as a dendrogram (tree)
- Each subtree corresponds to a cluster
- Height of branching shows distance between the objects
- There are two strategies: Agglomerative and Divisive

Hierarchical Clustering

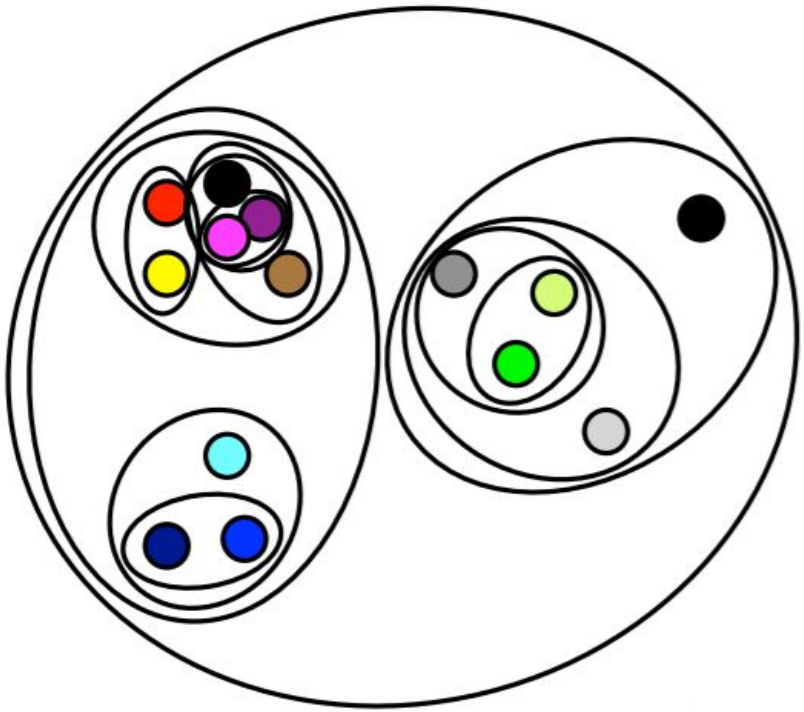
- Hierarchical clustering is usually visualized as a dendrogram (tree)
- Each subtree corresponds to a cluster
- Height of branching shows distance between the objects
- There are two strategies: Agglomerative and Divisive
  - **Agglomerative** clustering (Bottom-up): Start with a single-object clusters (singletons) and recursively merge them into larger clusters.

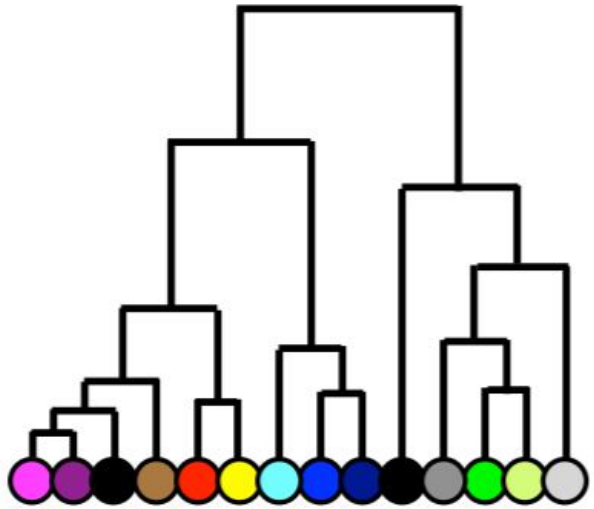
Hierarchical Clustering

- Hierarchical clustering is usually visualized as a dendrogram (tree)
- Each subtree corresponds to a cluster
- Height of branching shows distance between the objects
- There are two strategies: Agglomerative and Divisive
  - **Agglomerative** clustering (Bottom-up): Start with a single-object clusters (singletons) and recursively merge them into larger clusters.
  - **Divisive** clustering (Top down): Start with a cluster containing all data points and recursively divide it into smaller clusters.

Hierarchical Clustering

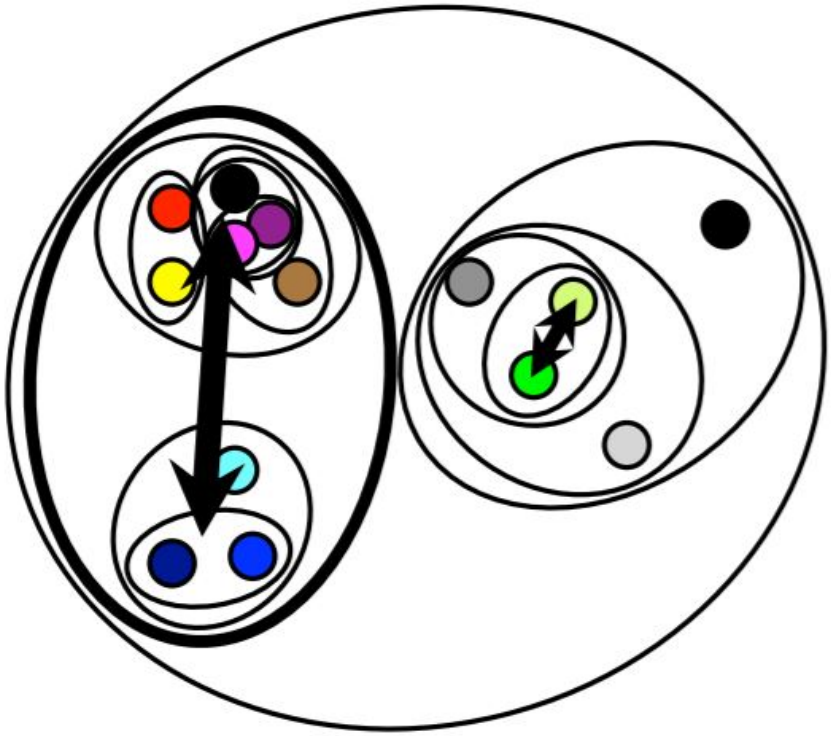
- Hierarchical clustering is usually visualized as a dendrogram (tree)

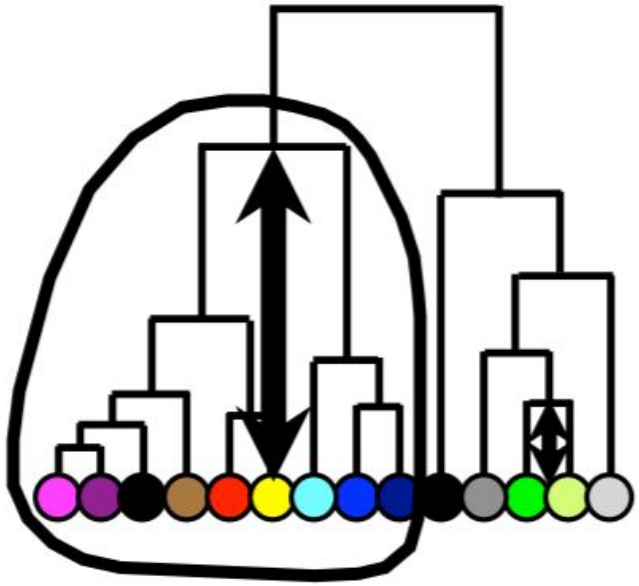




Hierarchical Clustering

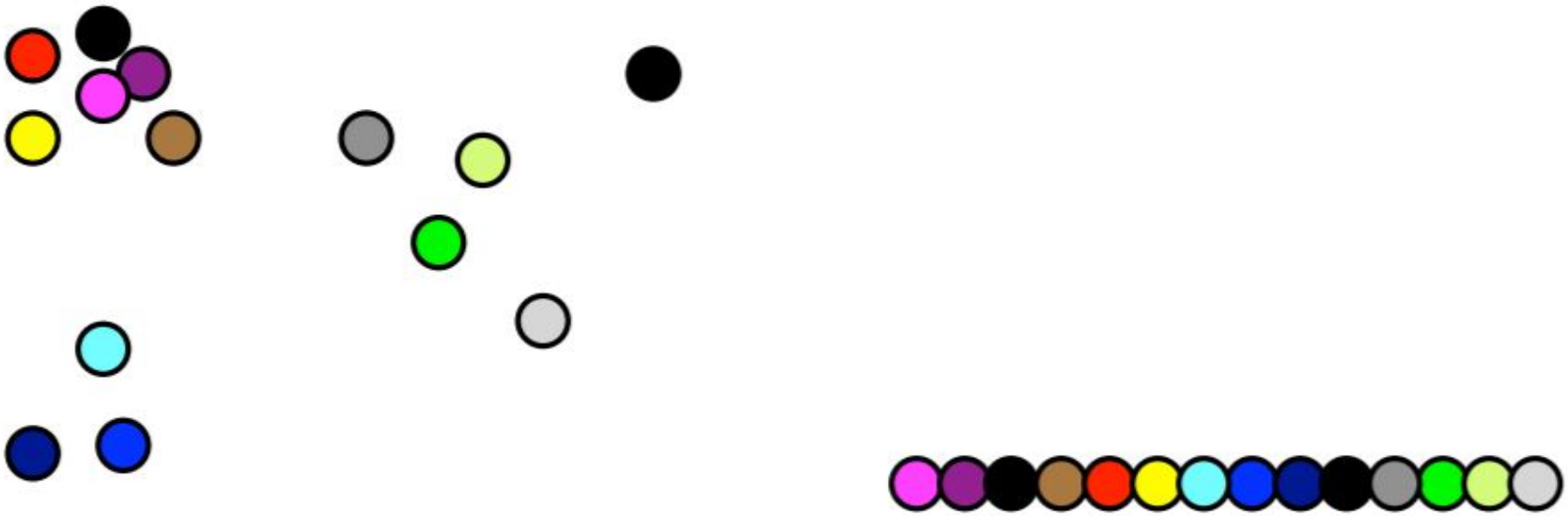
- Hierarchical clustering is usually visualized as a dendrogram (tree)





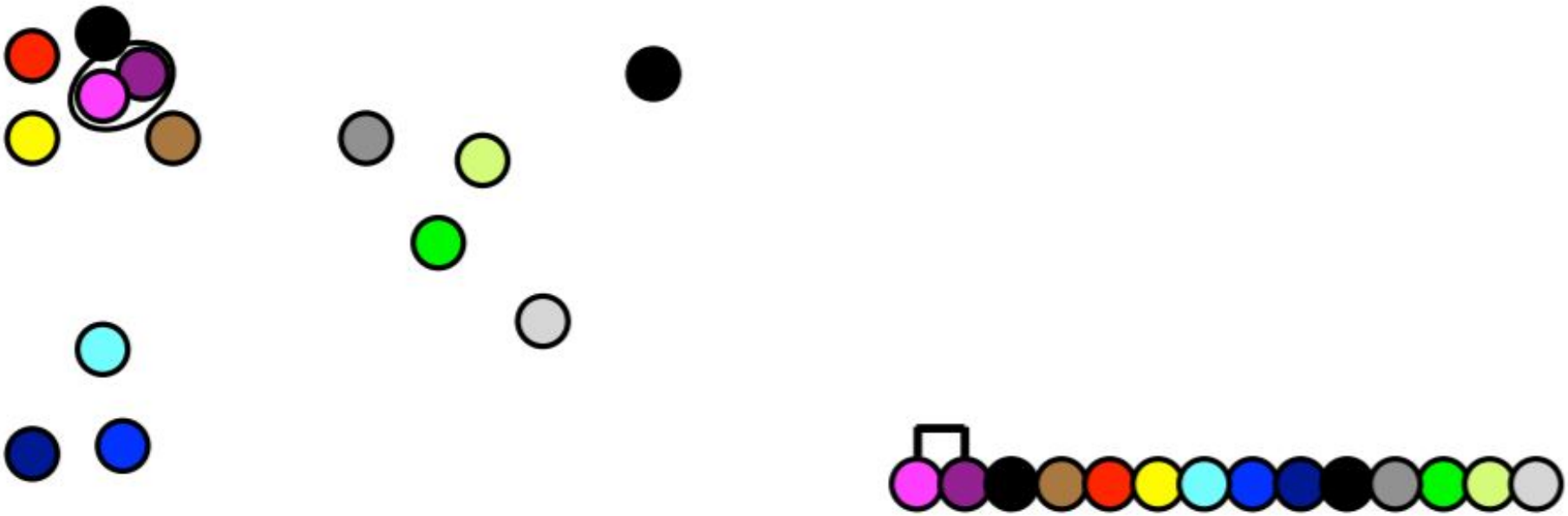
Hierarchical Clustering (Step 0)

Algorithm for Agglomerative Hierarchical Clustering: Join the two closest objects



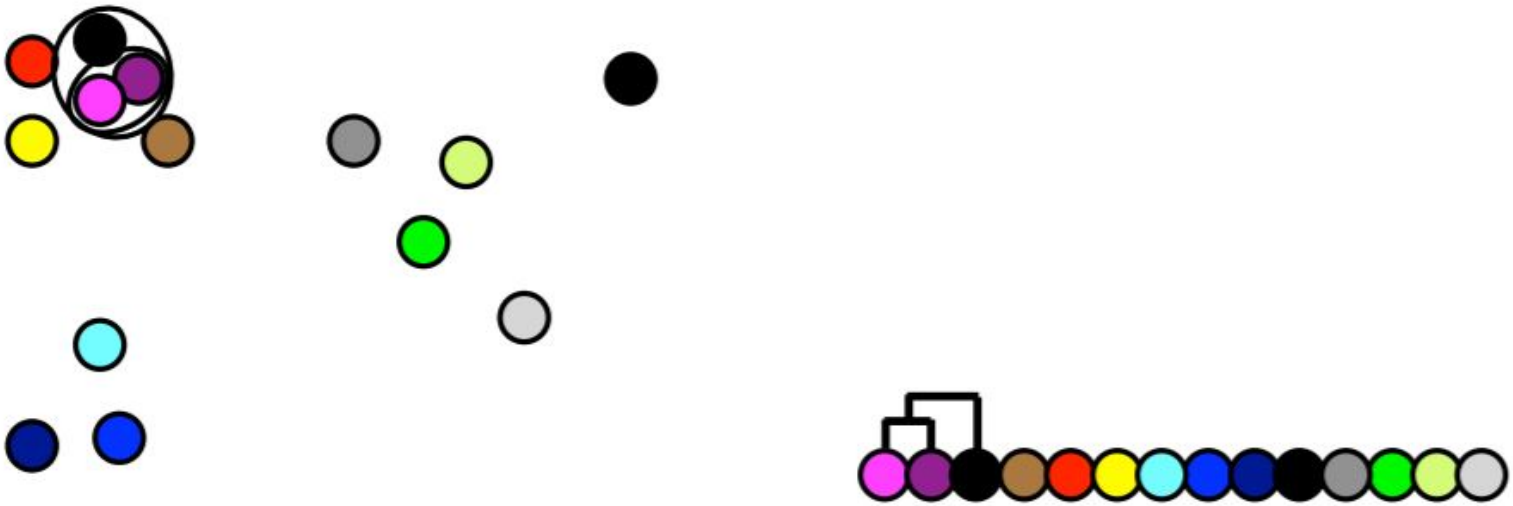
Hierarchical Clustering (Step 1)

Join the two closest objects (for example, closest with respect to Euclidean distance)



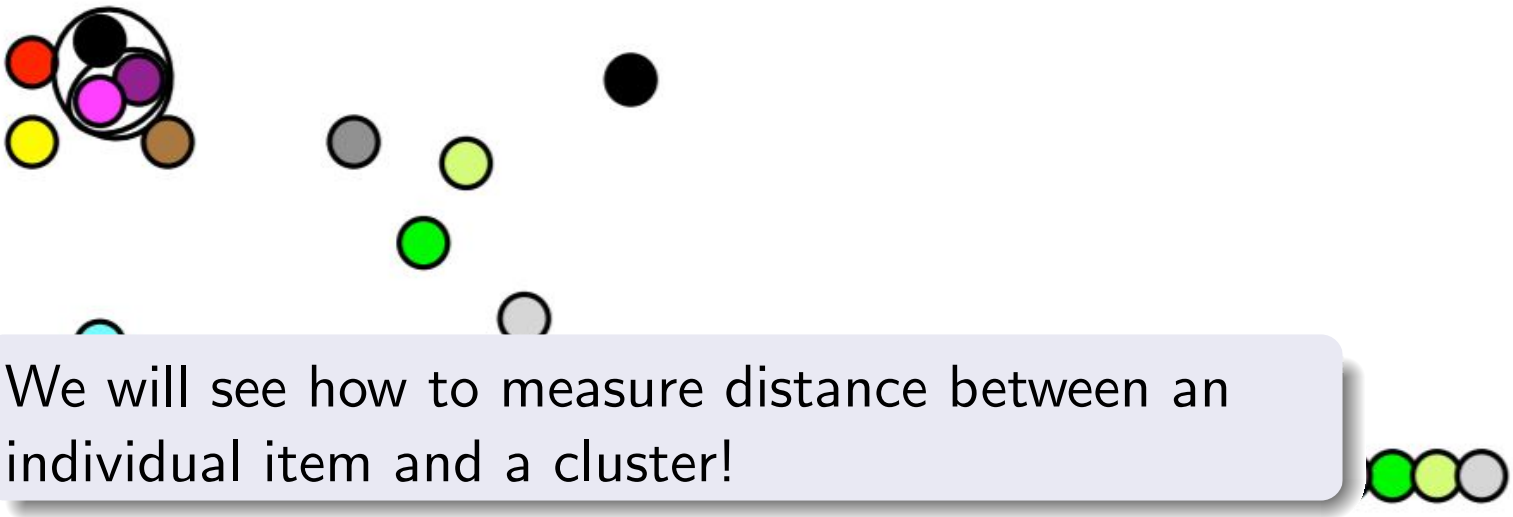
Hierarchical Clustering (Step 2)

Keep joining the closest pairs



Hierarchical Clustering (Step 2)

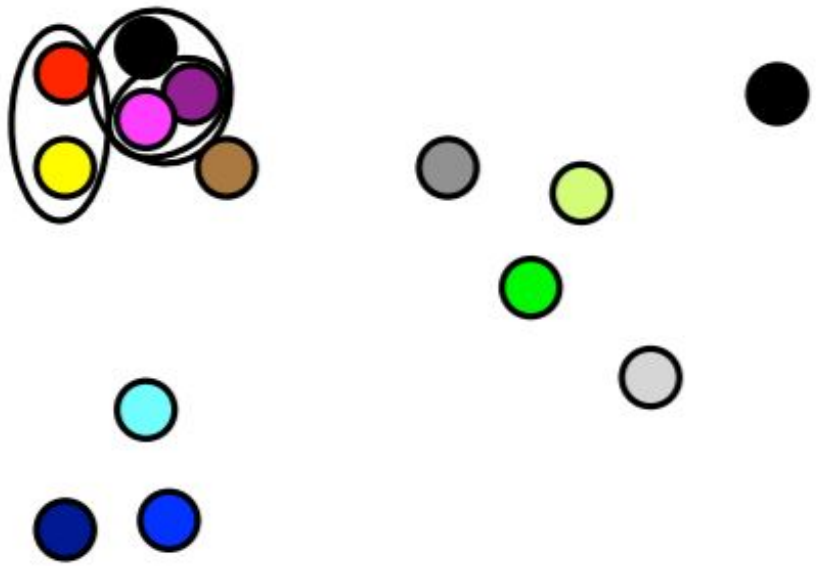
Keep joining the closest pairs



We will see how to measure distance between an individual item and a cluster!

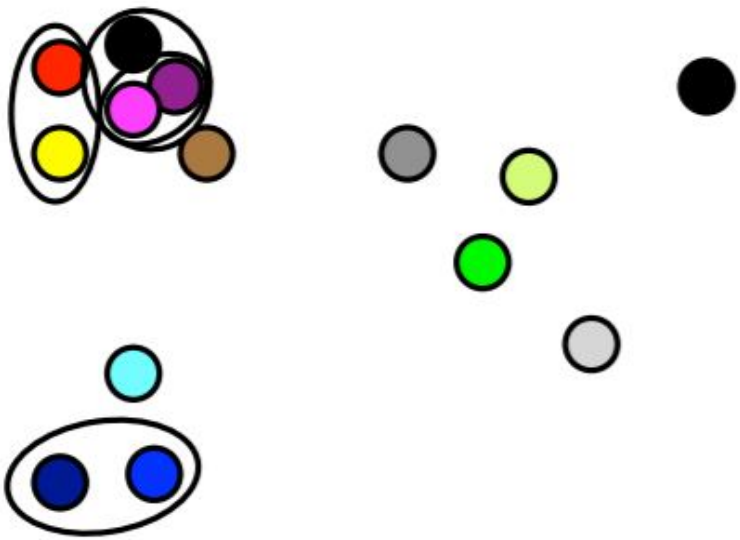
Hierarchical Clustering (Step 3)

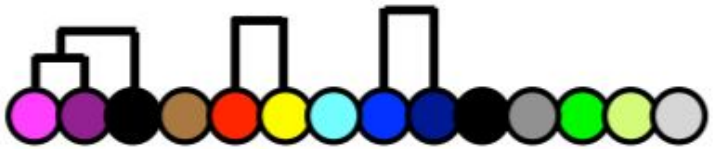
Keep joining the closest pairs



Hierarchical Clustering (Step 4)

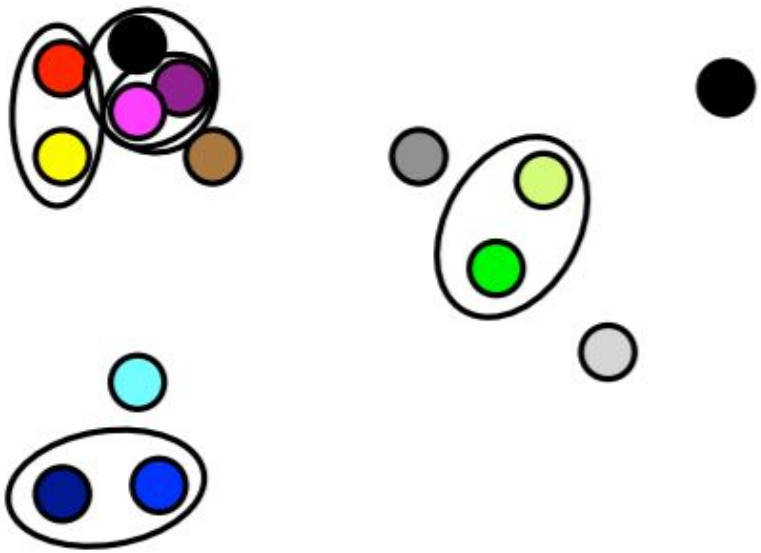
Keep joining the closest pairs

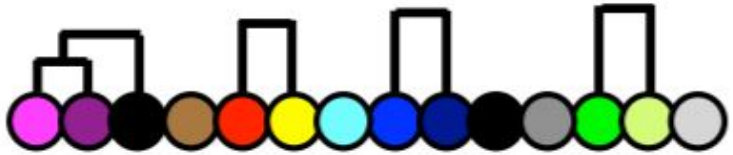




Hierarchical Clustering (Step 5)

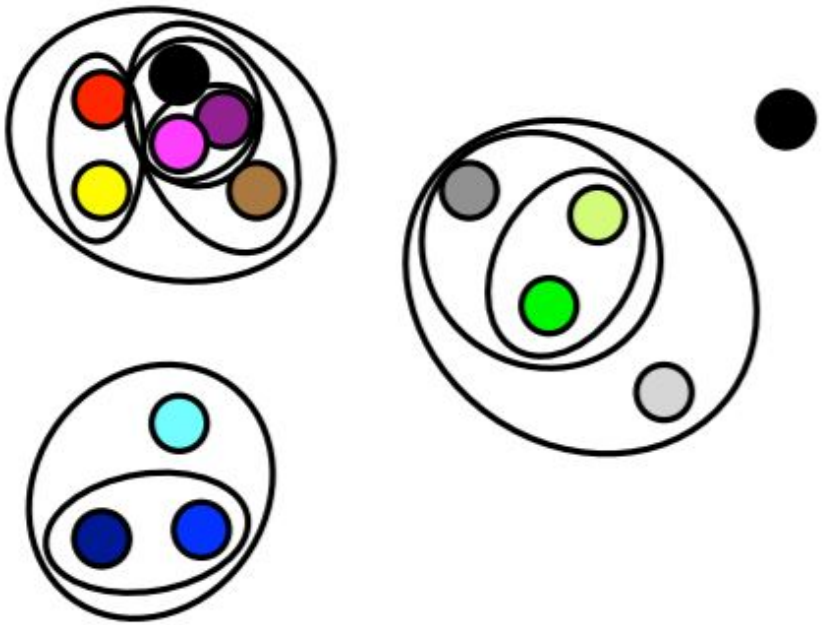
Keep joining the closest pairs

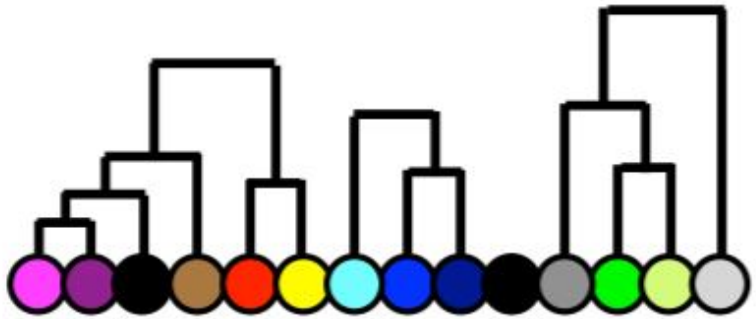




Hierarchical Clustering (Step 10)

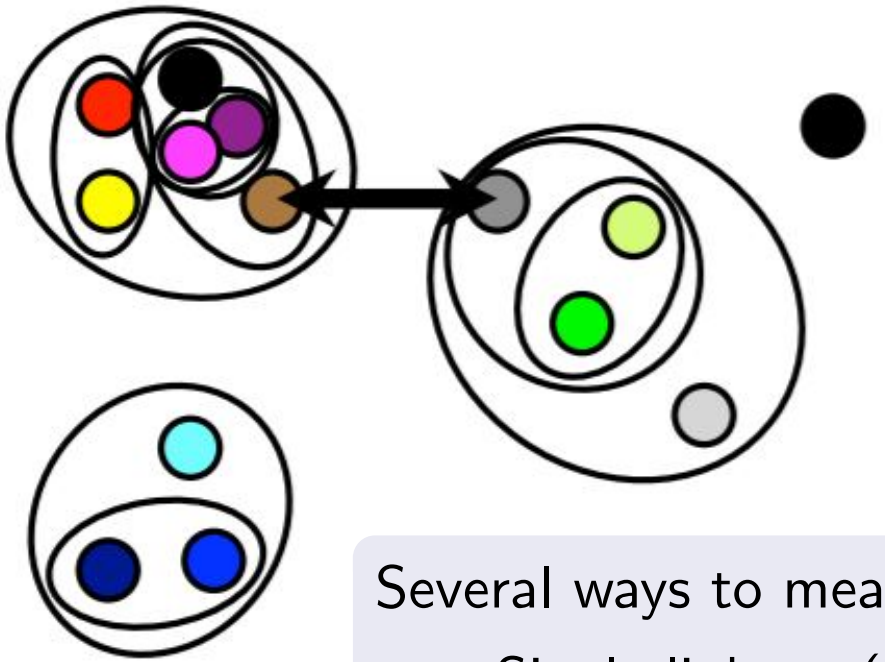
After 10 steps we have 4 clusters left





Hierarchical Clustering (Step 10)

After 10 steps we have 4 clusters left

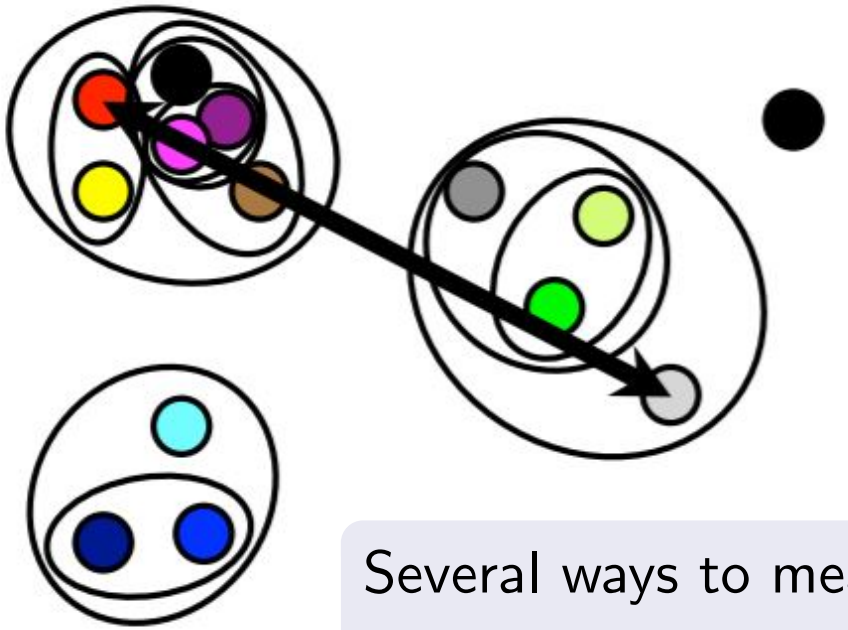


Several ways to measure distance between clusters:

- Single linkage (a.k.a. nearest neighbour technique)

Hierarchical Clustering (Step 10)

After 10 steps we have 4 clusters left

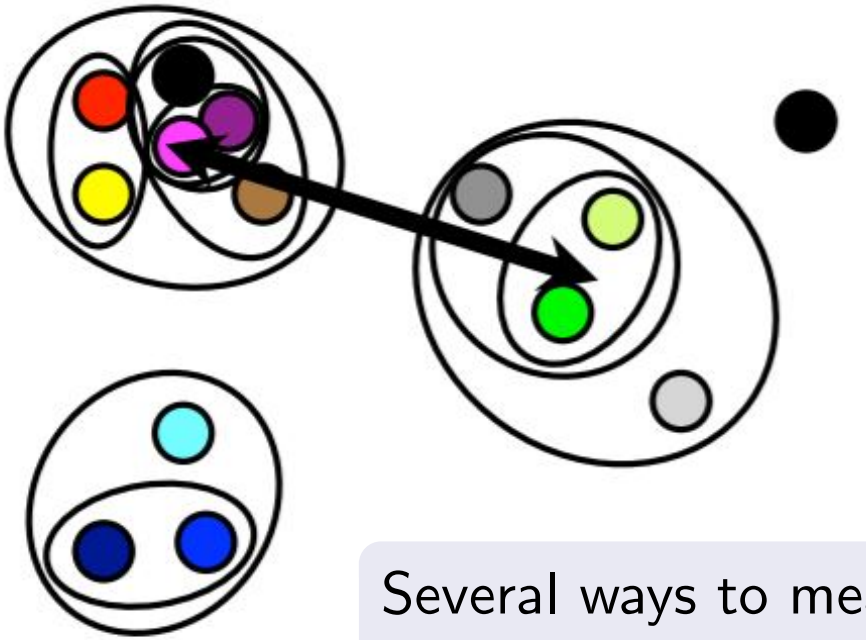


Several ways to measure distance between clusters:

- Single linkage (a.k.a. nearest neighbour technique)
- Complete linkage (a.k.a furthest neighbour technique)

Hierarchical Clustering (Step 10)

After 10 steps we have 4 clusters left

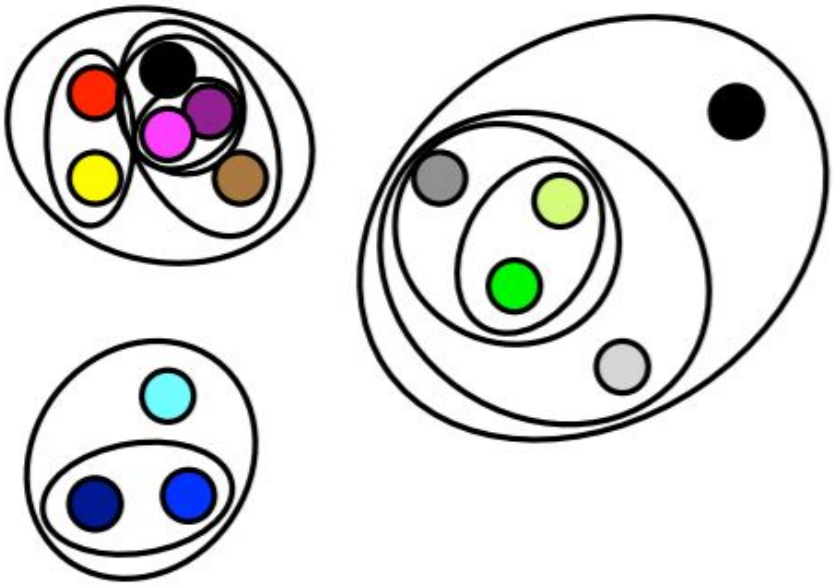


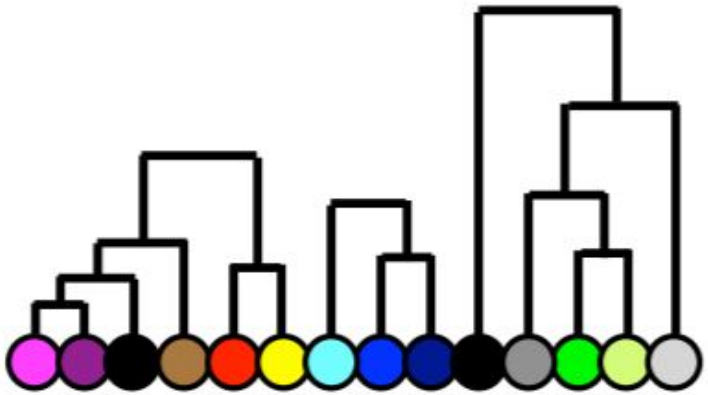
Several ways to measure distance between clusters:

- Single linkage (a.k.a. nearest neighbour technique)
- Complete linkage (a.k.a furthest neighbour technique)
- Average linkage
  -Weighted
  -Unweighted

Hierarchical Clustering (Step 11)

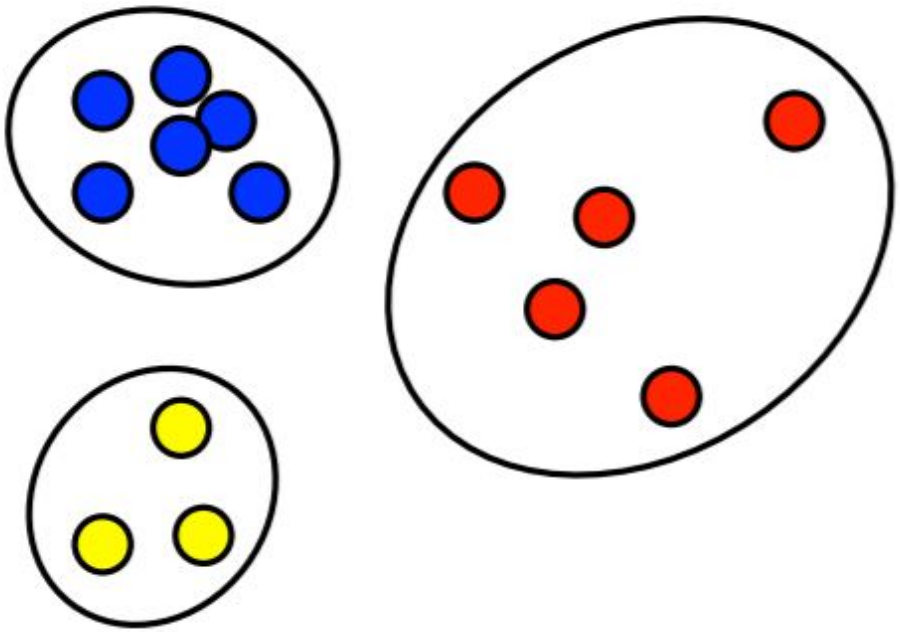
In this example and at this stage we have the same result as in our partitional clustering example

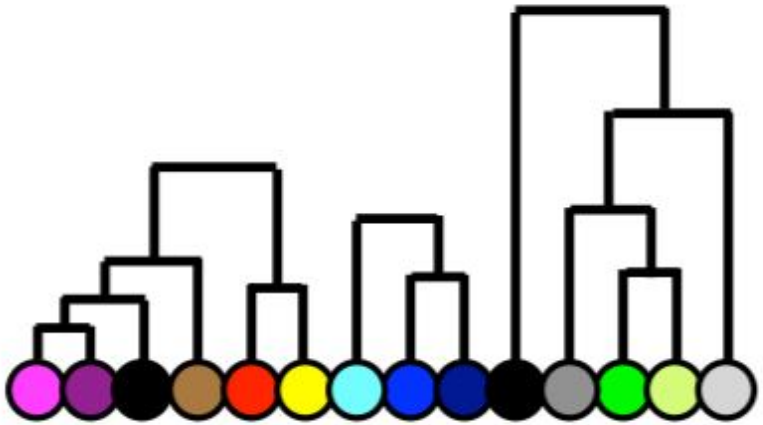




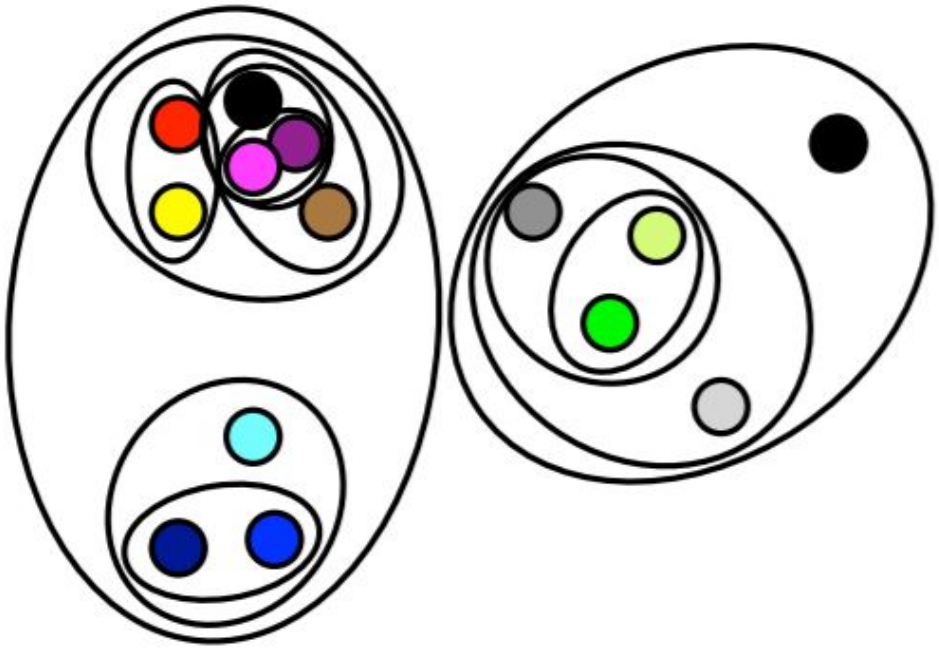
Hierarchical Clustering (Step 11)

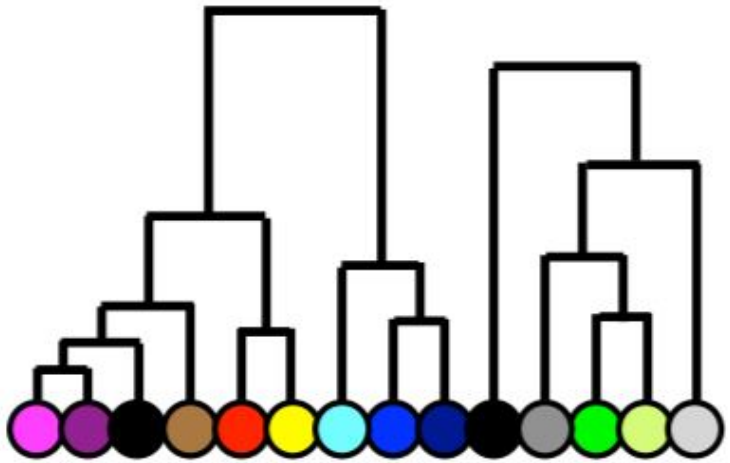
In this example and at this stage we have the same result as in our partitional clustering example





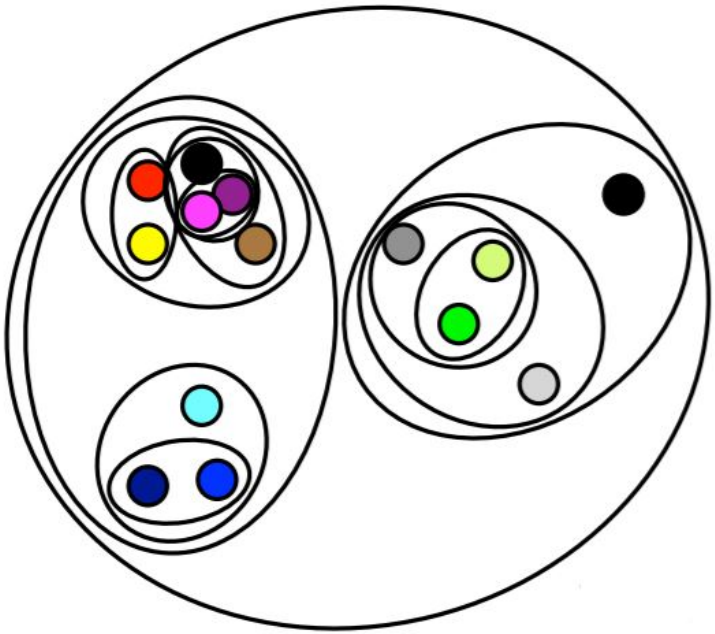
Hierarchical Clustering (Step 12)

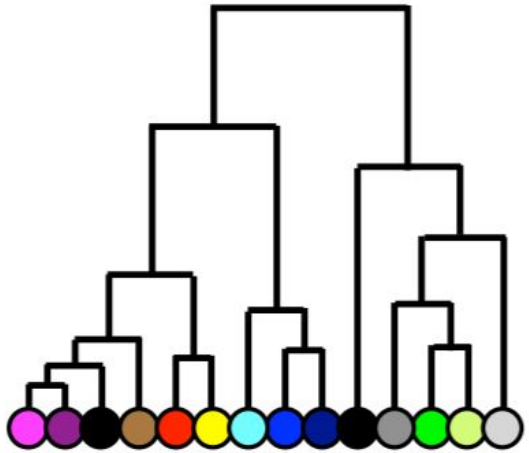




Hierarchical Clustering (Step 13)

In the final step the two remaining clusters are joined into a single cluster





Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).

Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).
  - Suffers from *chaining* meaning that clusters can be too spread out, and not compact enough

Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).
  - Suffers from *chaining* meaning that clusters can be too spread out, and not compact enough
  - Clusters can violate the "compactness" property that all observations within each cluster tend to be similar to one another.

Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).
  - Suffers from *chaining* meaning that clusters can be too spread out, and not compact enough
  - Clusters can violate the "compactness" property that all observations within each cluster tend to be similar to one another.
- **Complete** Linkage - distance is computed between the two LEAST similar parts of clusters (two most distant points).

Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).
  - Suffers from *chaining* meaning that clusters can be too spread out, and not compact enough
  - Clusters can violate the "compactness" property that all observations within each cluster tend to be similar to one another.
- **Complete** Linkage - distance is computed between the two LEAST similar parts of clusters (two most distant points).
  - Avoids chaining, but suffers from *crowding*, meaning that clusters are compact, but not far enough apart

Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).
  - Suffers from *chaining* meaning that clusters can be too spread out, and not compact enough
  - Clusters can violate the "compactness" property that all observations within each cluster tend to be similar to one another.
- **Complete** Linkage - distance is computed between the two LEAST similar parts of clusters (two most distant points).
  - Avoids chaining, but suffers from *crowding*, meaning that clusters are compact, but not far enough apart
  - Observations assigned to a cluster can be much closer to members of other clusters than they are to some members of their own cluster

Summary of Hierarchical Clustering

- **Single** Linkage - distance is computed between the two MOST similar parts of clusters (two closest points).
  - Suffers from *chaining* meaning that clusters can be too spread out, and not compact enough
  - Clusters can violate the "compactness" property that all observations within each cluster tend to be similar to one another.
- **Complete** Linkage - distance is computed between the two LEAST similar parts of clusters (two most distant points).
  - Avoids chaining, but suffers from *crowding*, meaning that clusters are compact, but not far enough apart
  - Observations assigned to a cluster can be much closer to members of other clusters than they are to some members of their own cluster
- **Average** Linkage - distance is computed between clusters' centroids. This is a balanced approach: clusters tend to be relatively compact and relatively far apart.

Summary of Hierarchical Clustering

Pro: The clusters can be visualized and interpreted with dendrograms

Summary of Hierarchical Clustering

Pro: The clusters can be visualized and interpreted with dendrograms

Pro: The number of clusters is not pre-defined and can be chosen after inspecting the dendrogram

Summary of Hierarchical Clustering

Pro: The clusters can be visualized and interpreted with dendrograms

Pro: The number of clusters is not pre-defined and can be chosen after inspecting the dendrogram

Pro: In some problems, hierarchy of clusters is preferred over flat clusters

Summary of Hierarchical Clustering

Pro: The clusters can be visualized and interpreted with dendrograms

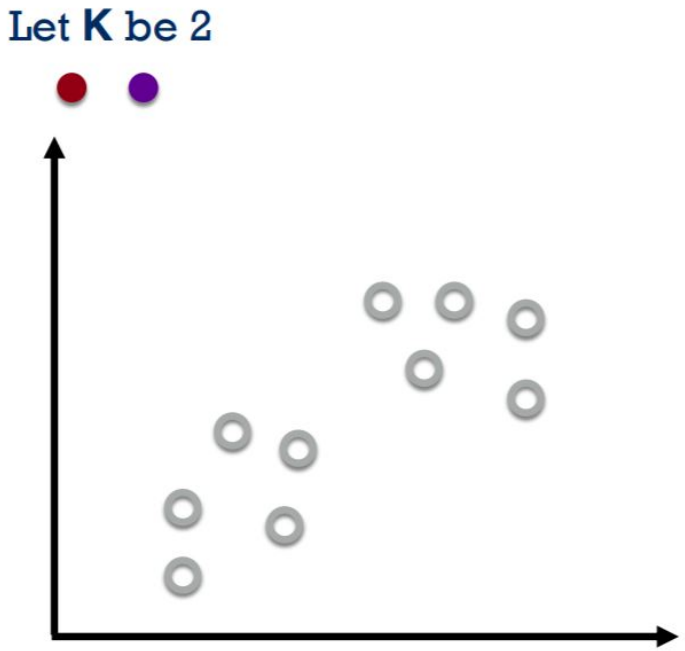
Pro: The number of clusters is not pre-defined and can be chosen after inspecting the dendrogram

Pro: In some problems, hierarchy of clusters is preferred over flat clusters

Con: Hierarchical clustering can't handle big data well, because of its quadratic time complexity

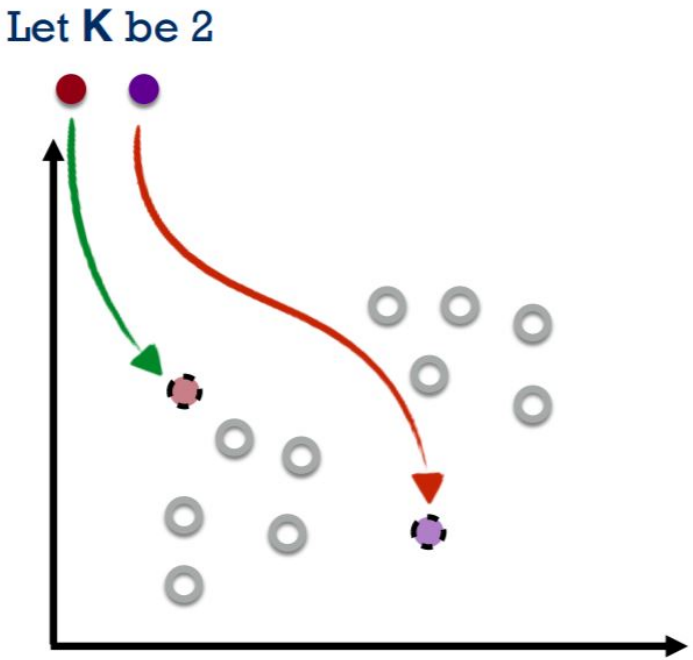
K-means Clustering

1. Choose K, the number of potential clusters



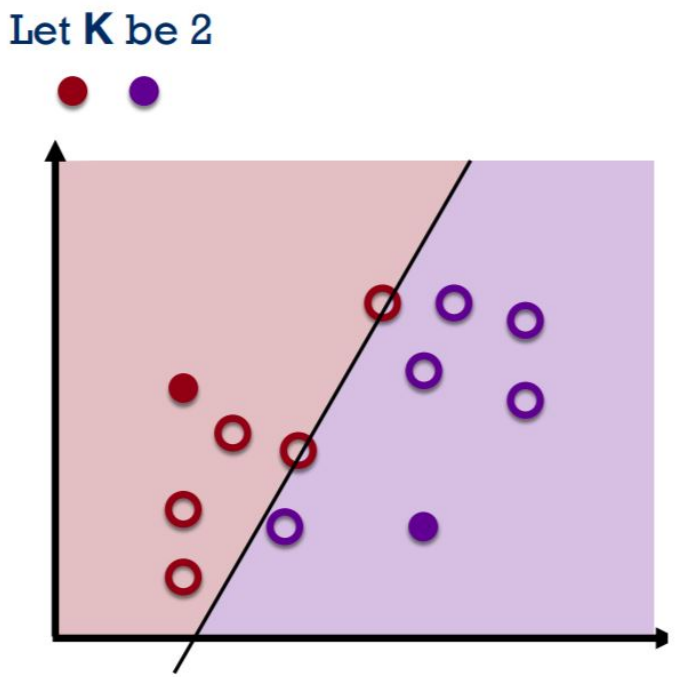
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data



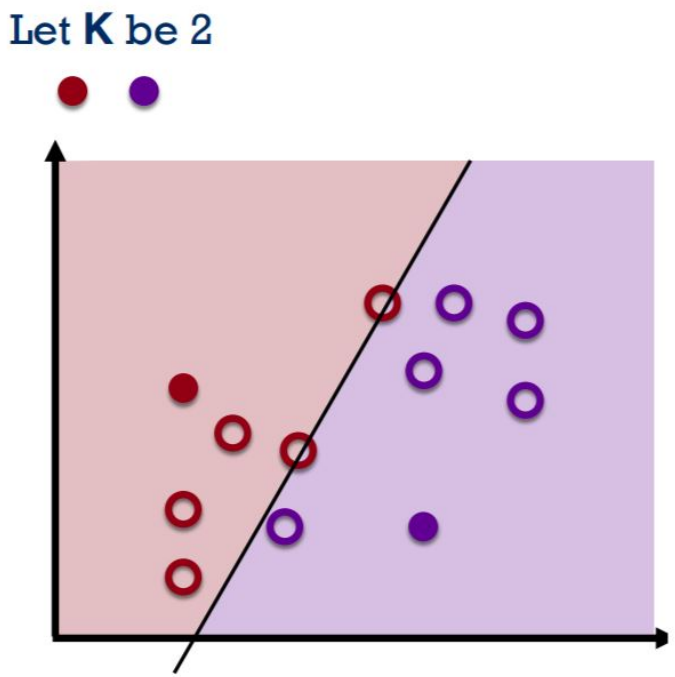
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre



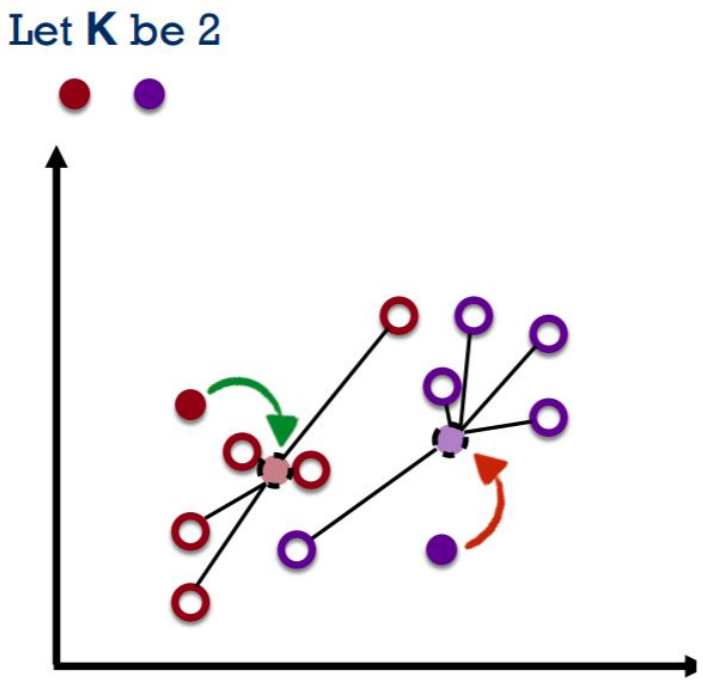
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre



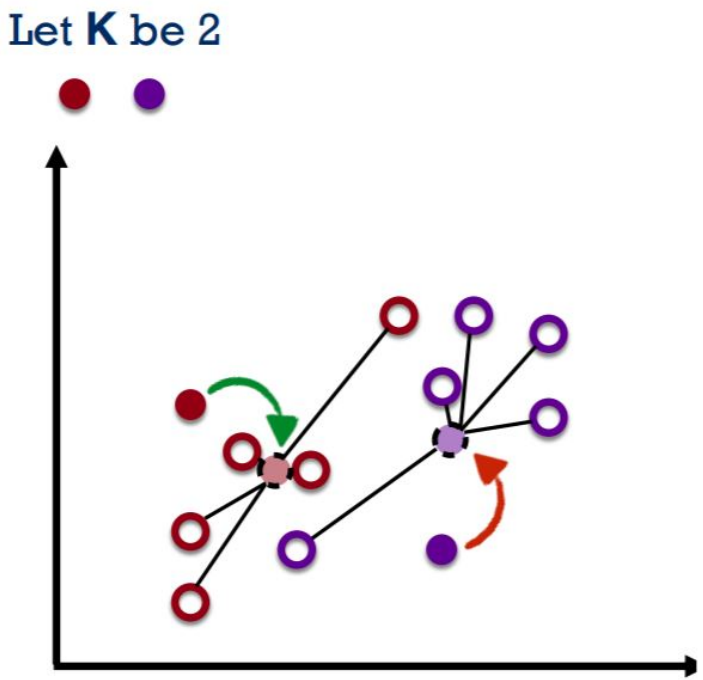
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers



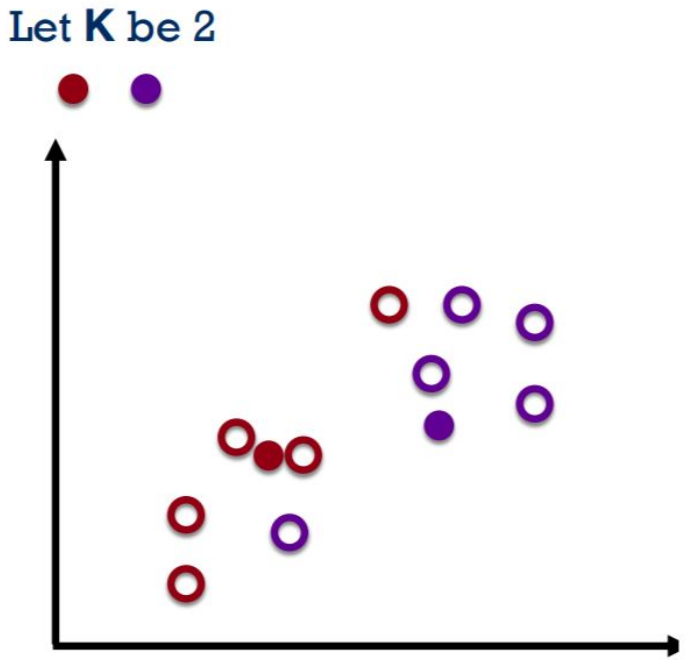
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers



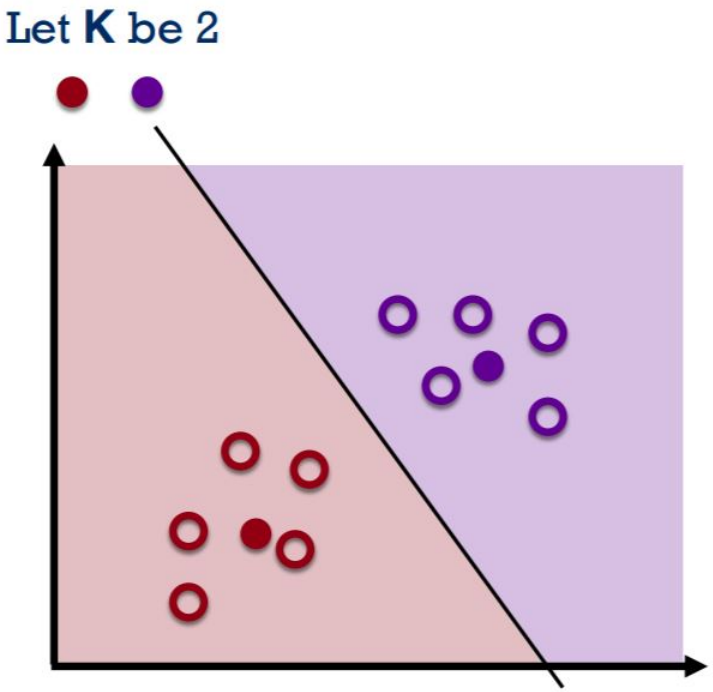
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers
5. Steps 3 and 4 are repeated until convergence

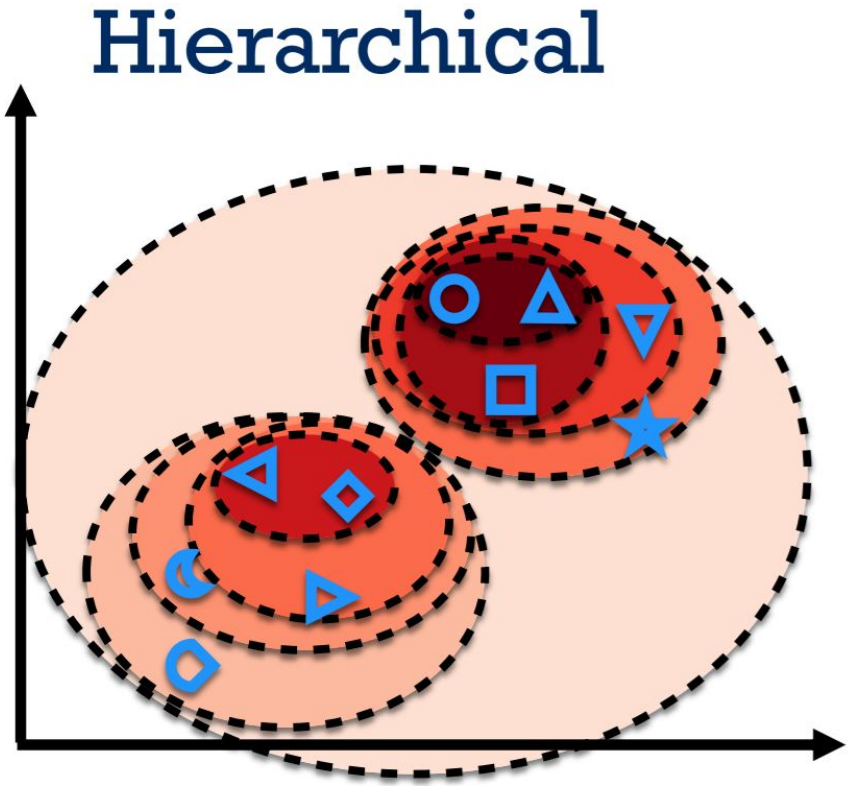


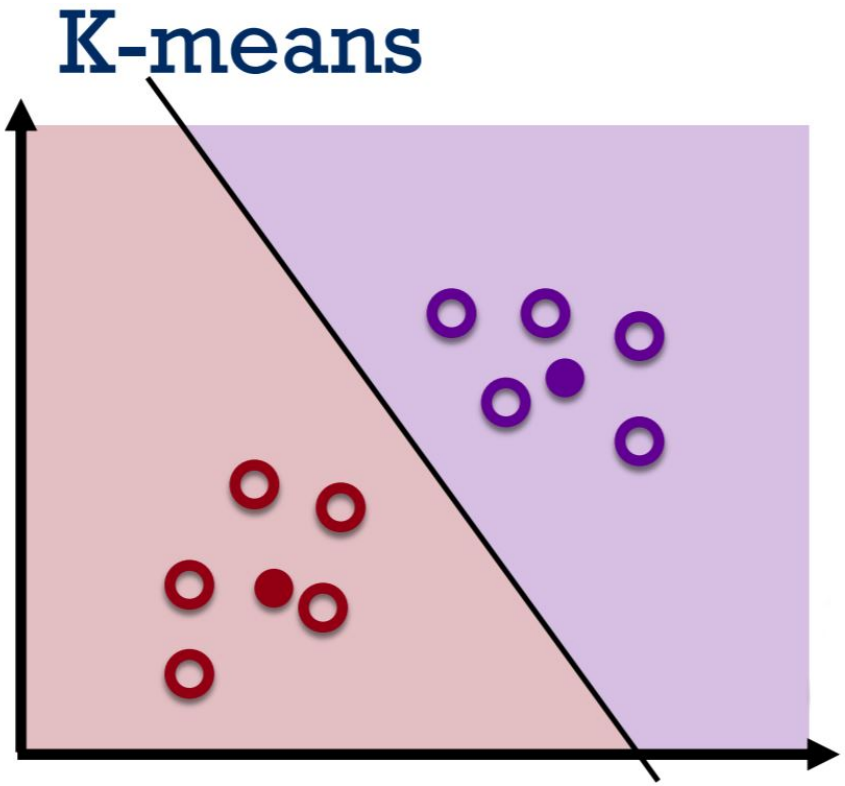
K-means Clustering

1. Choose K, the number of potential clusters
2. Initialise cluster centers randomly within the data
3. Instances are clustered to the nearest (Euclidean distance) cluster centre
4. Centroids of each of the K clusters become new cluster centers
5. Steps 3 and 4 are repeated until convergence



Hierarchical vs K-means





How to choose K in K-means?

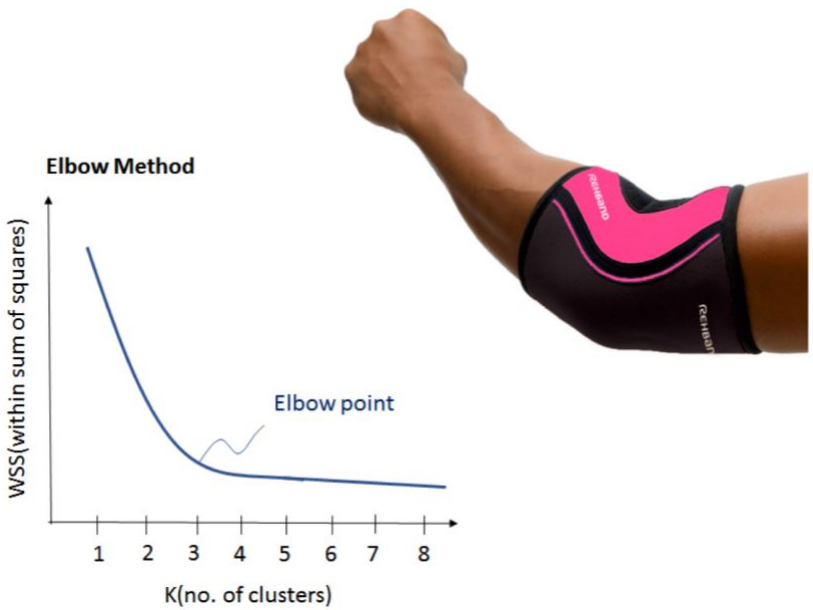
- **Elbow method:** increase K until it does not help to describe data better

How to choose K in K-means?

- **Elbow method:** increase K until it does not help to describe data better
- We are interested in finding K such that the sum of within-group Euclidean distances is smaller

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} \|\mathbf{x}_i^{(j)} - \mathbf{c}_j\|^2,$$

where $\mathbf{c}_j$ is the centroid (mean) of the j^{th} cluster



Summary of K-means

Pro: Simple, easy to implement

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Con: The dissimilarity measure is fixed to Euclidean distance and features should be quantitative (numeric)

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Con: The dissimilarity measure is fixed to Euclidean distance and features should be quantitative (numeric)

Con: Squared Euclidean distance places the highest influence on the largest distances, therefore this approach is sensitive to outliers in the data

Summary of K-means

Pro: Simple, easy to implement

Pro: Easy to interpret the clustering results;

Pro: Fast and efficient in terms of computational cost

Con: The number of clusters (K) needs to be defined in advance

Con: The dissimilarity measure is fixed to Euclidean distance and features should be quantitative (numeric)

Con: Squared Euclidean distance places the highest influence on the largest distances, therefore this approach is sensitive to outliers in the data

Con: In K Means clustering, the results produced by running the algorithm multiple times might differ because of the random initialization of the centroids. While results are reproducible in Hierarchical clustering.

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure

K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure
4. Medoids of each of the K clusters are updated, taking the ones that are closer to all other points in the cluster

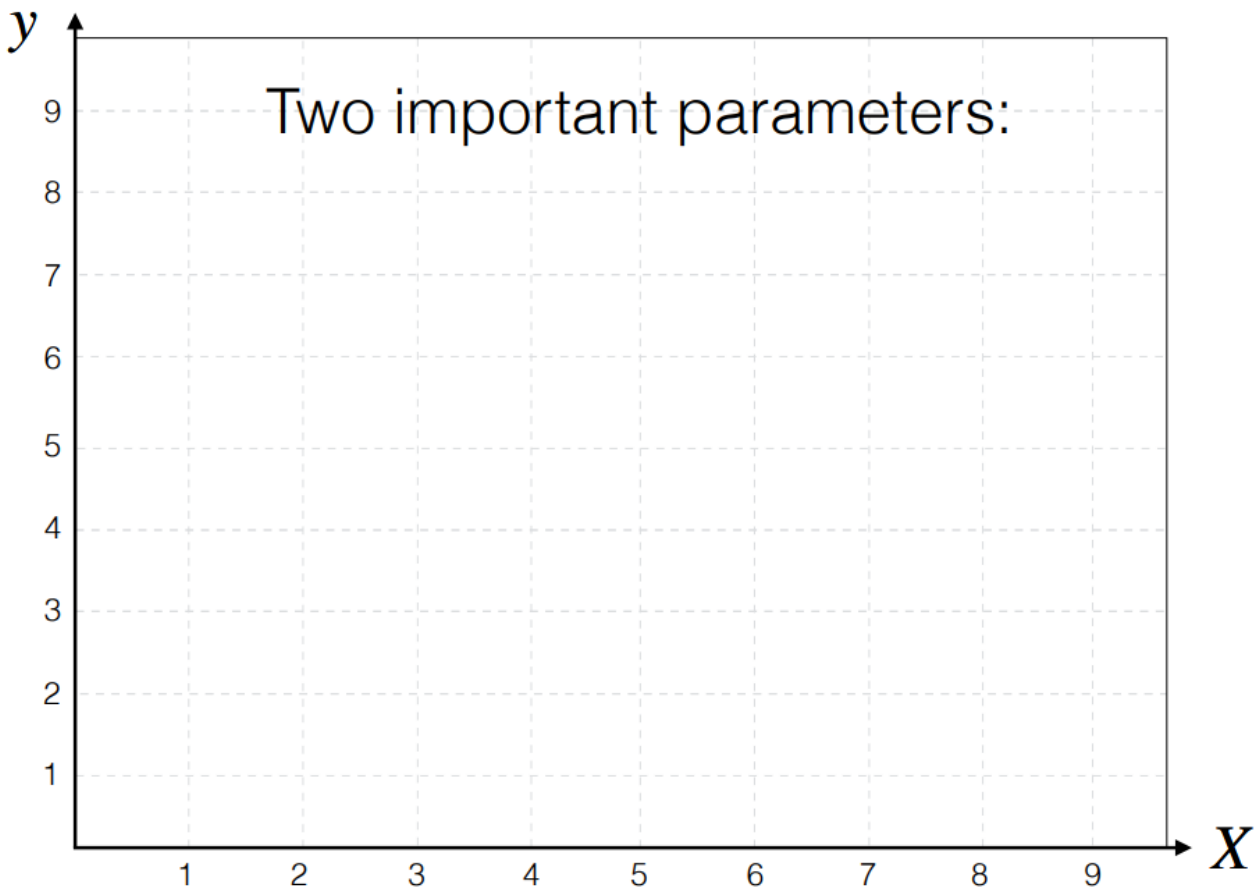
K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure
4. Medoids of each of the K clusters are updated, taking the ones that are closer to all other points in the cluster

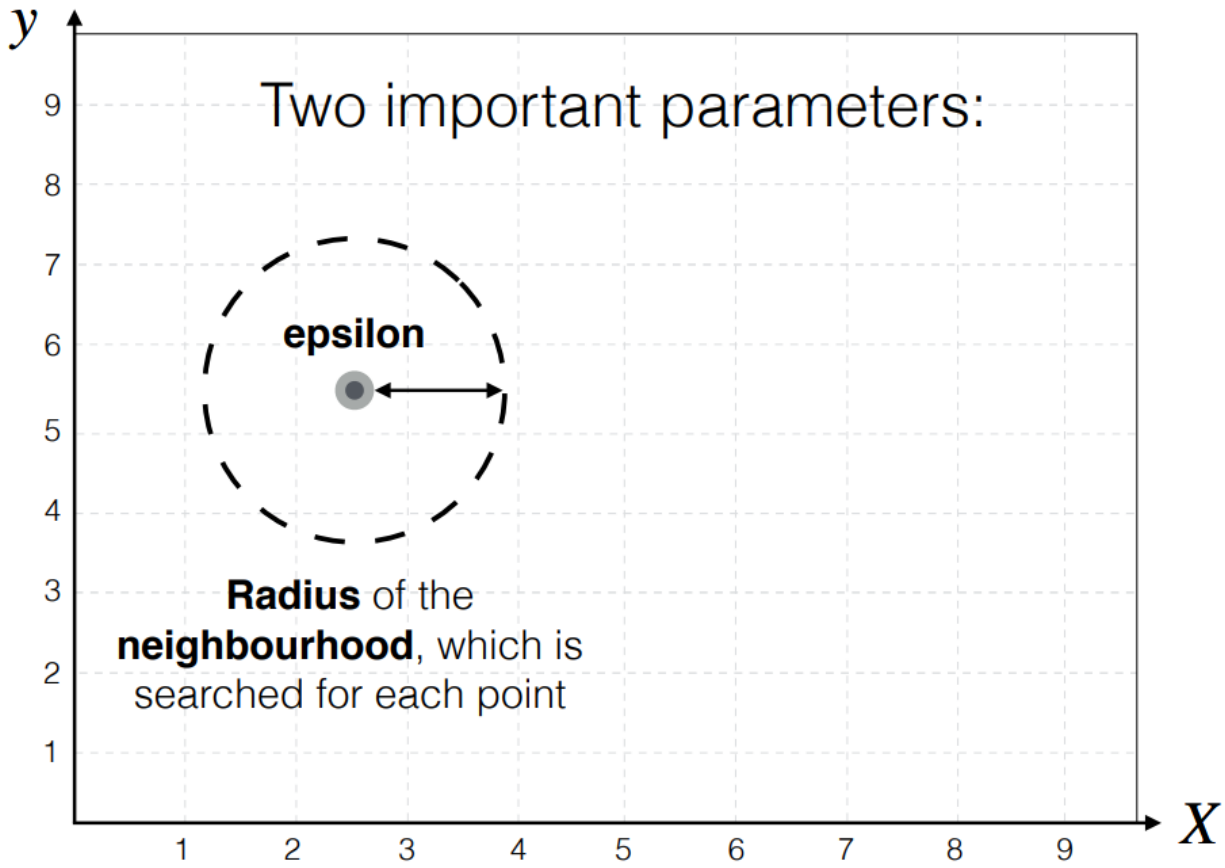
K-medoids (Partitioning Around Medoids) clustering

1. Choose K, the number of potential clusters
2. Initialise cluster medoids (central points) randomly within the data
3. Instances are clustered to the nearest cluster medoid according to a predefined dissimilarity measure
4. Medoids of each of the K clusters are updated, taking the ones that are closer to all other points in the cluster
5. Steps 3 and 4 are repeated until convergence

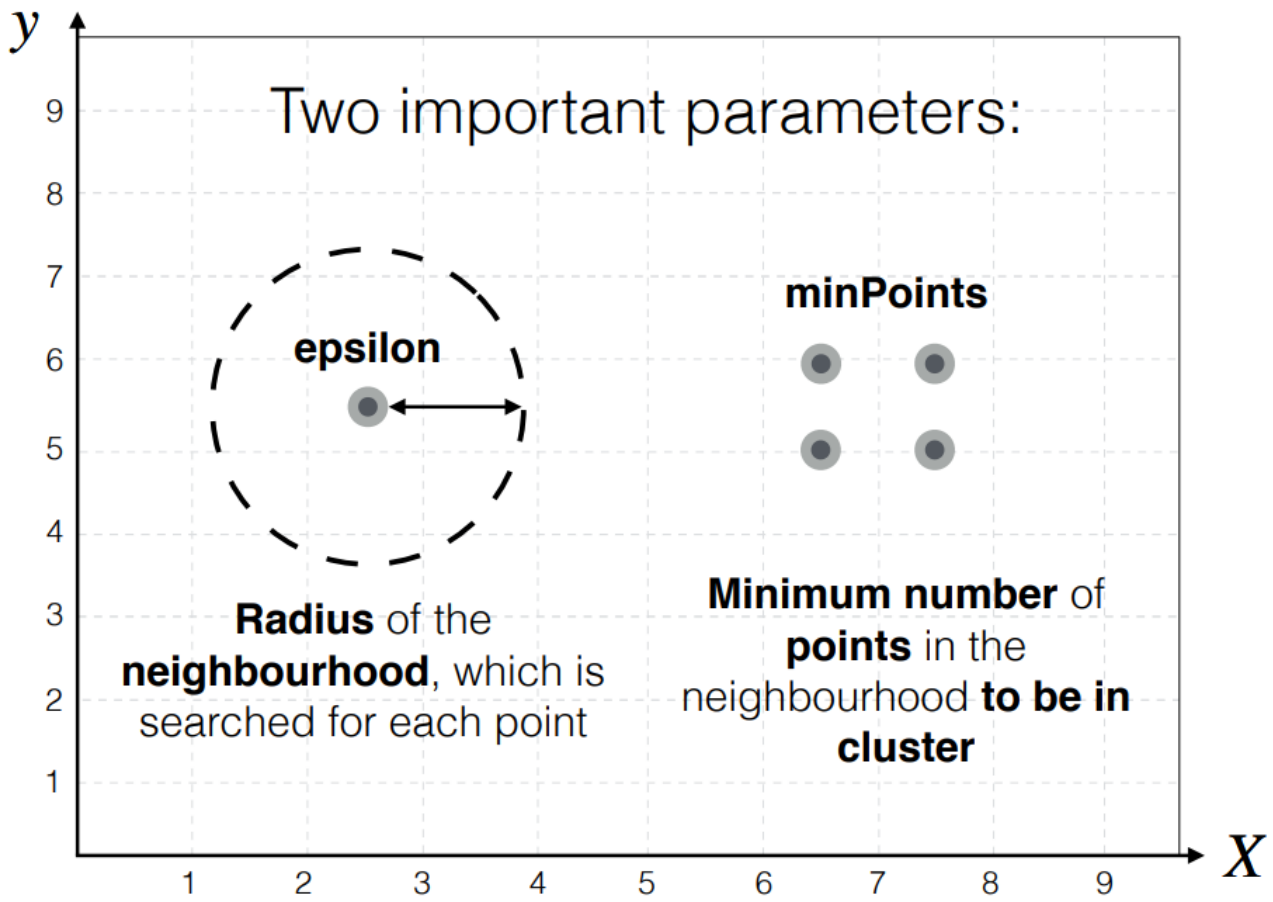
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



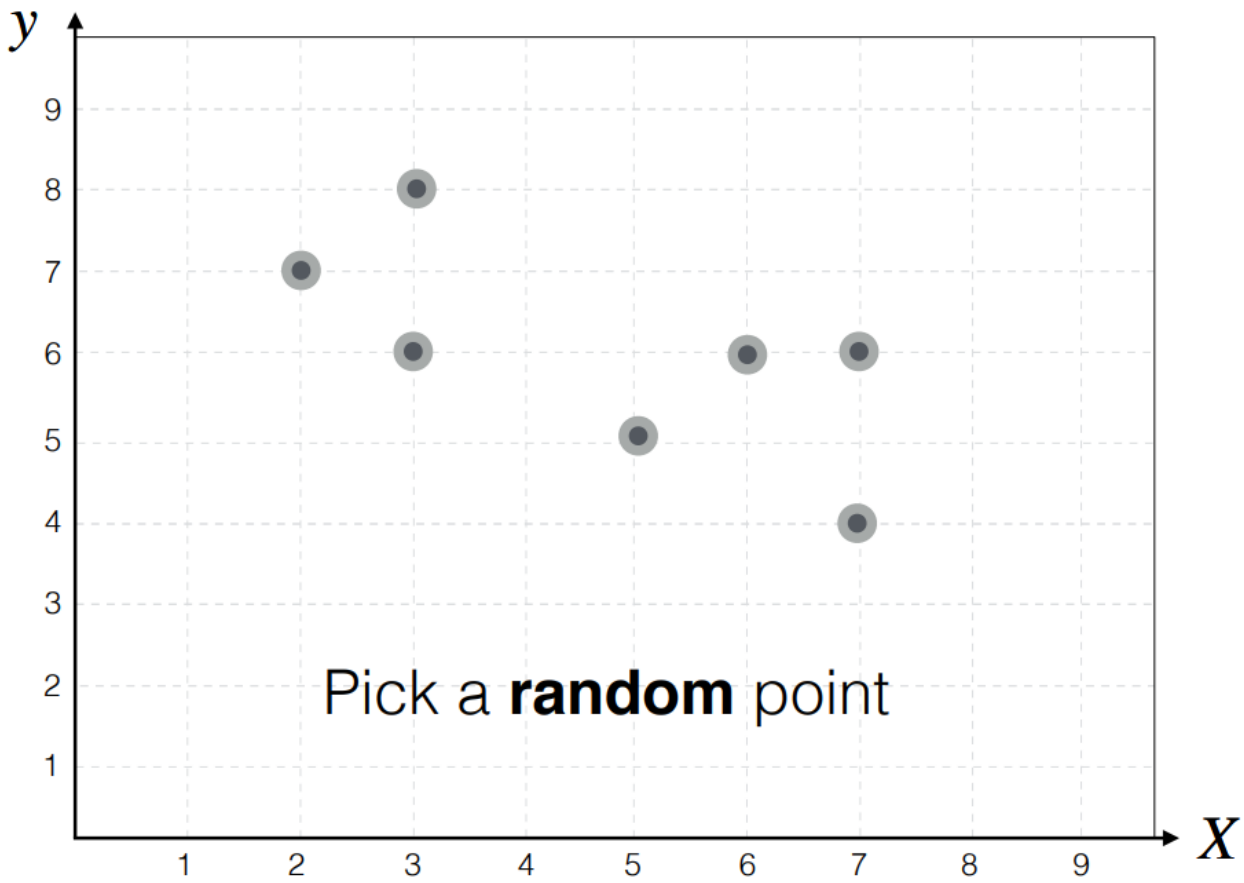
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



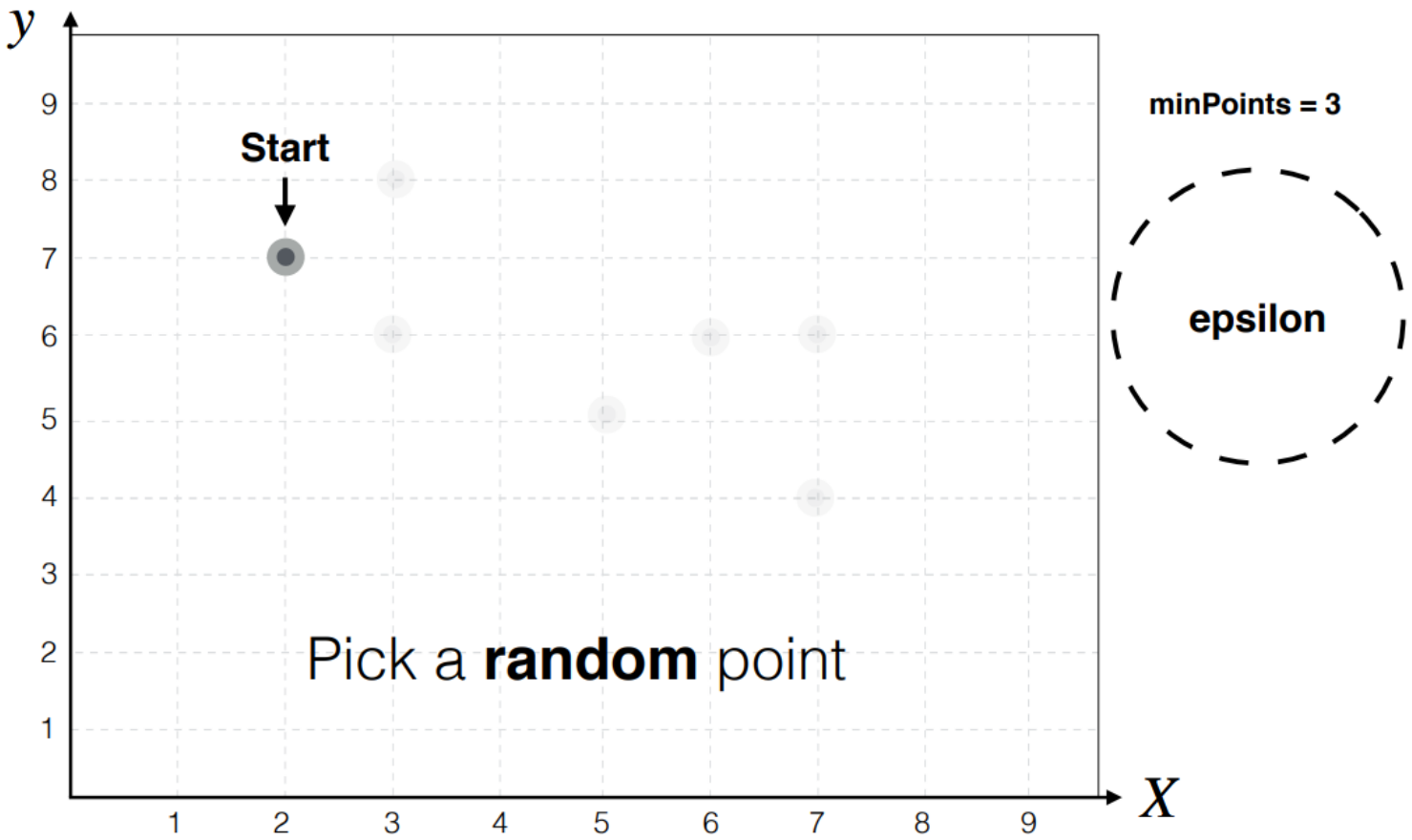
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



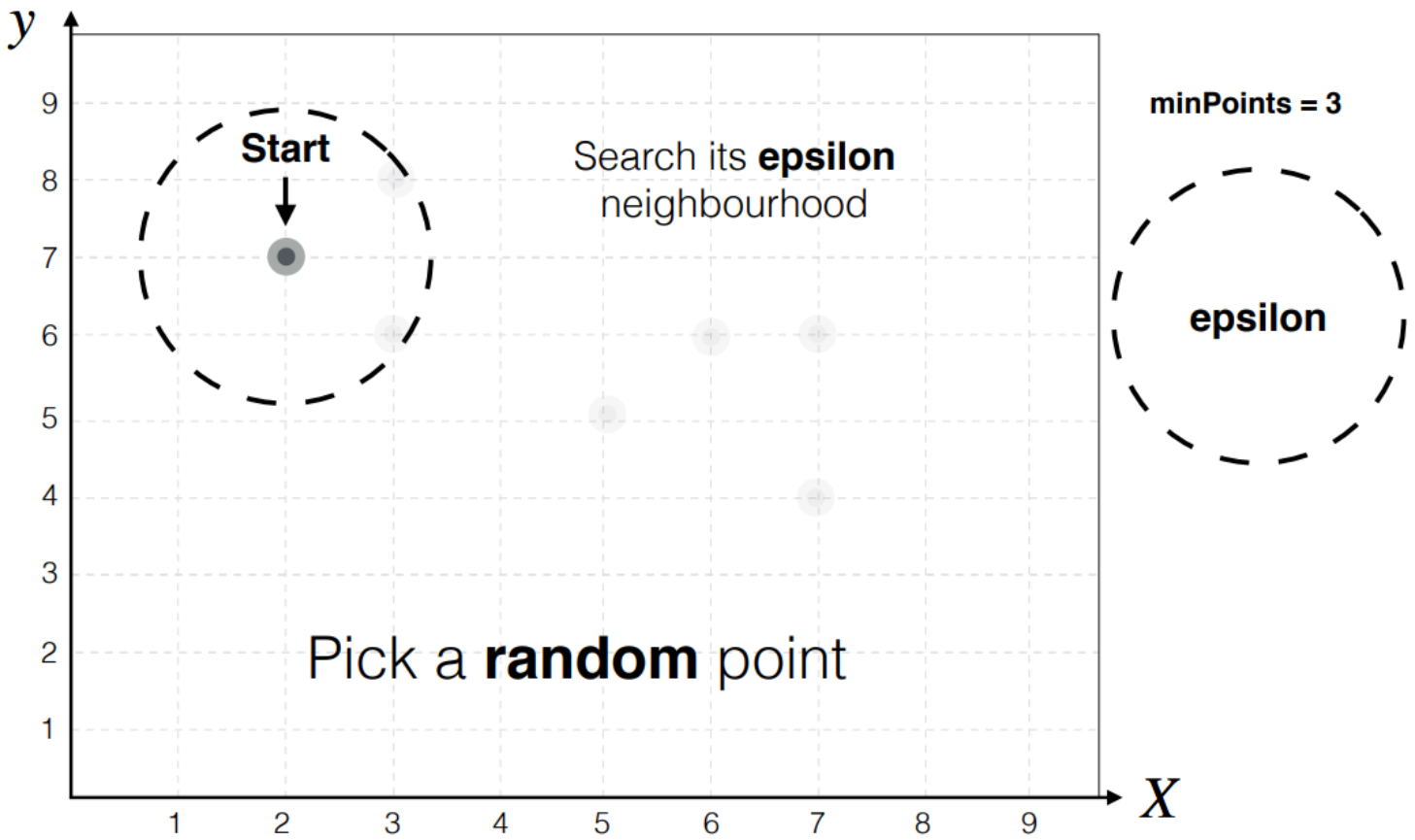
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



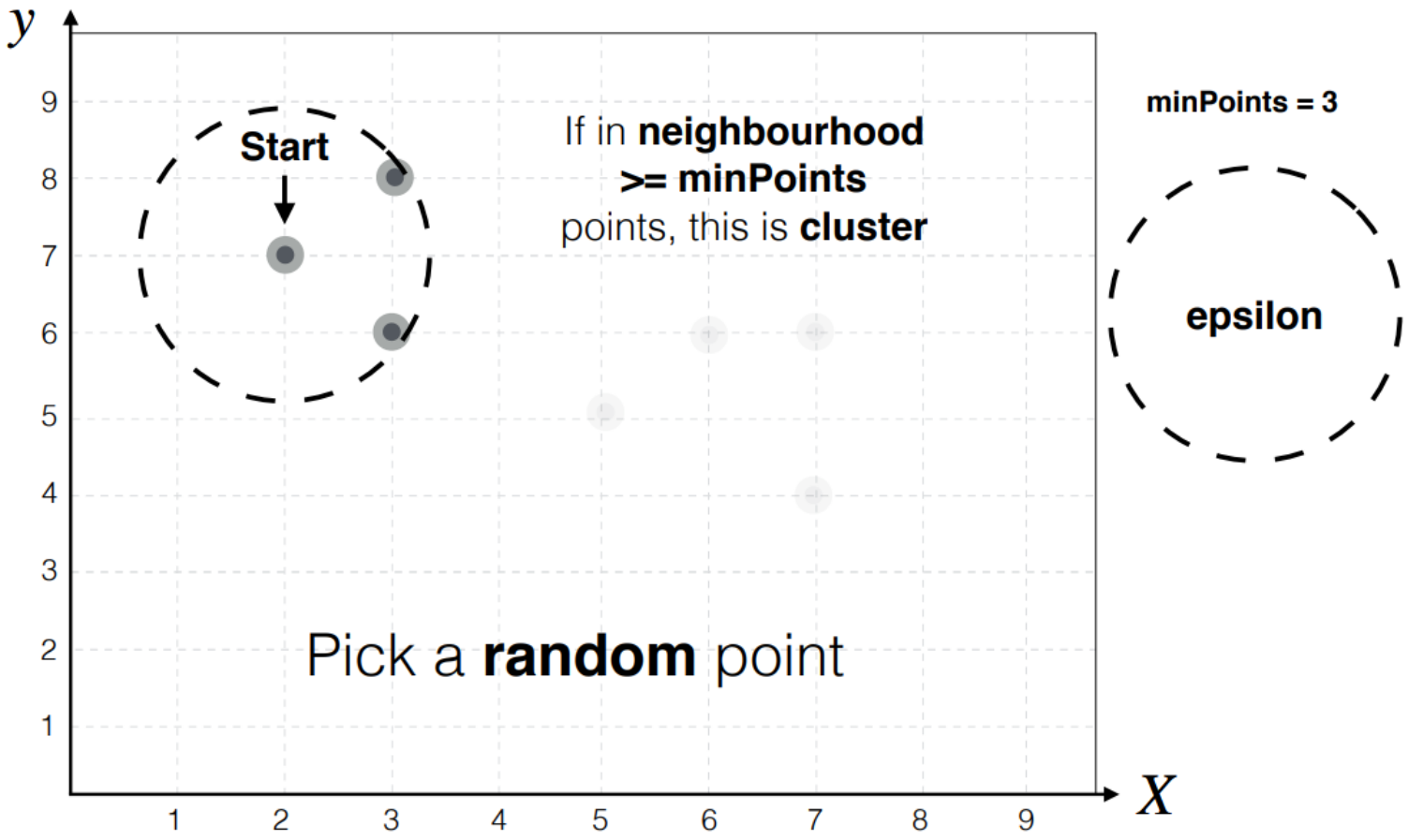
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



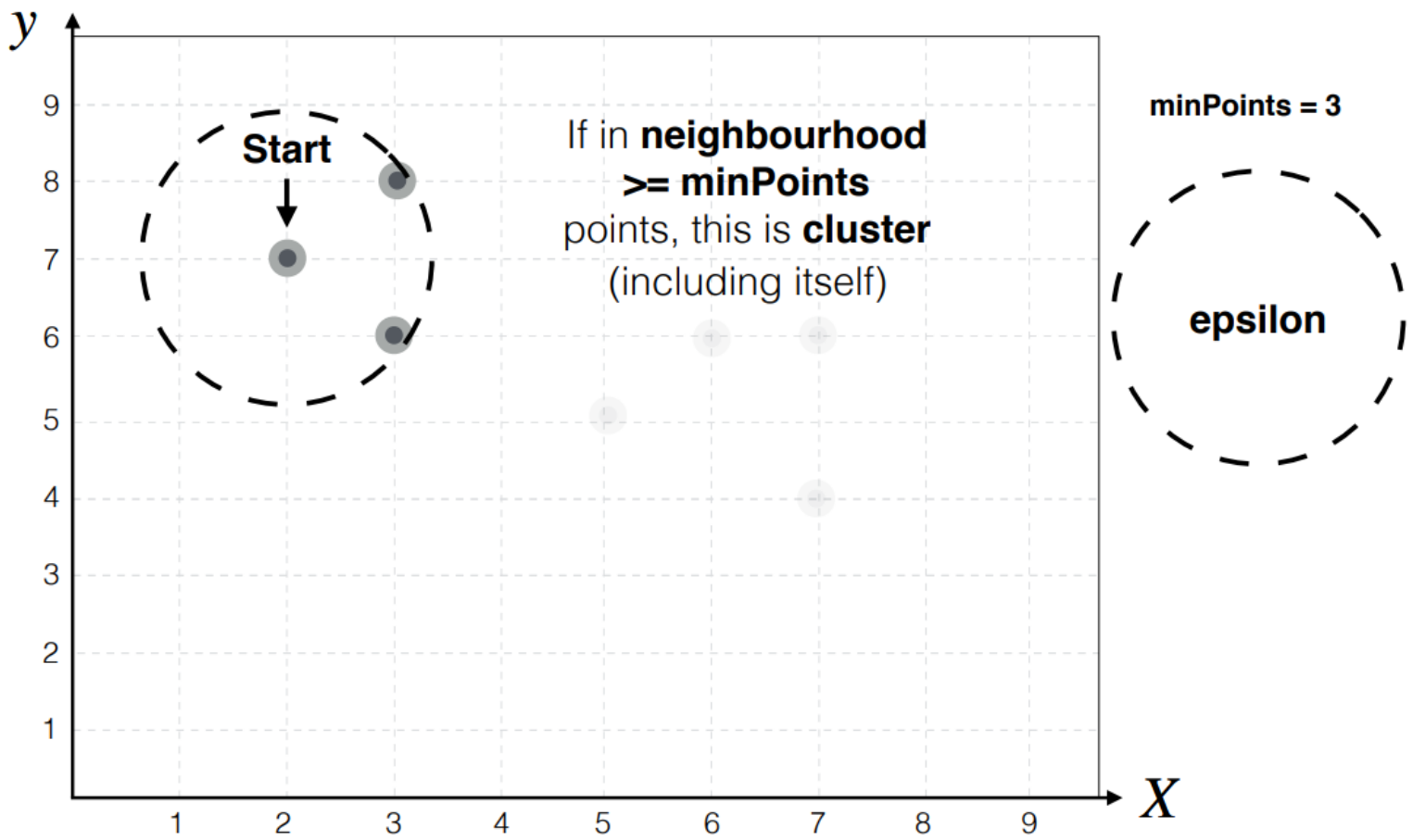
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



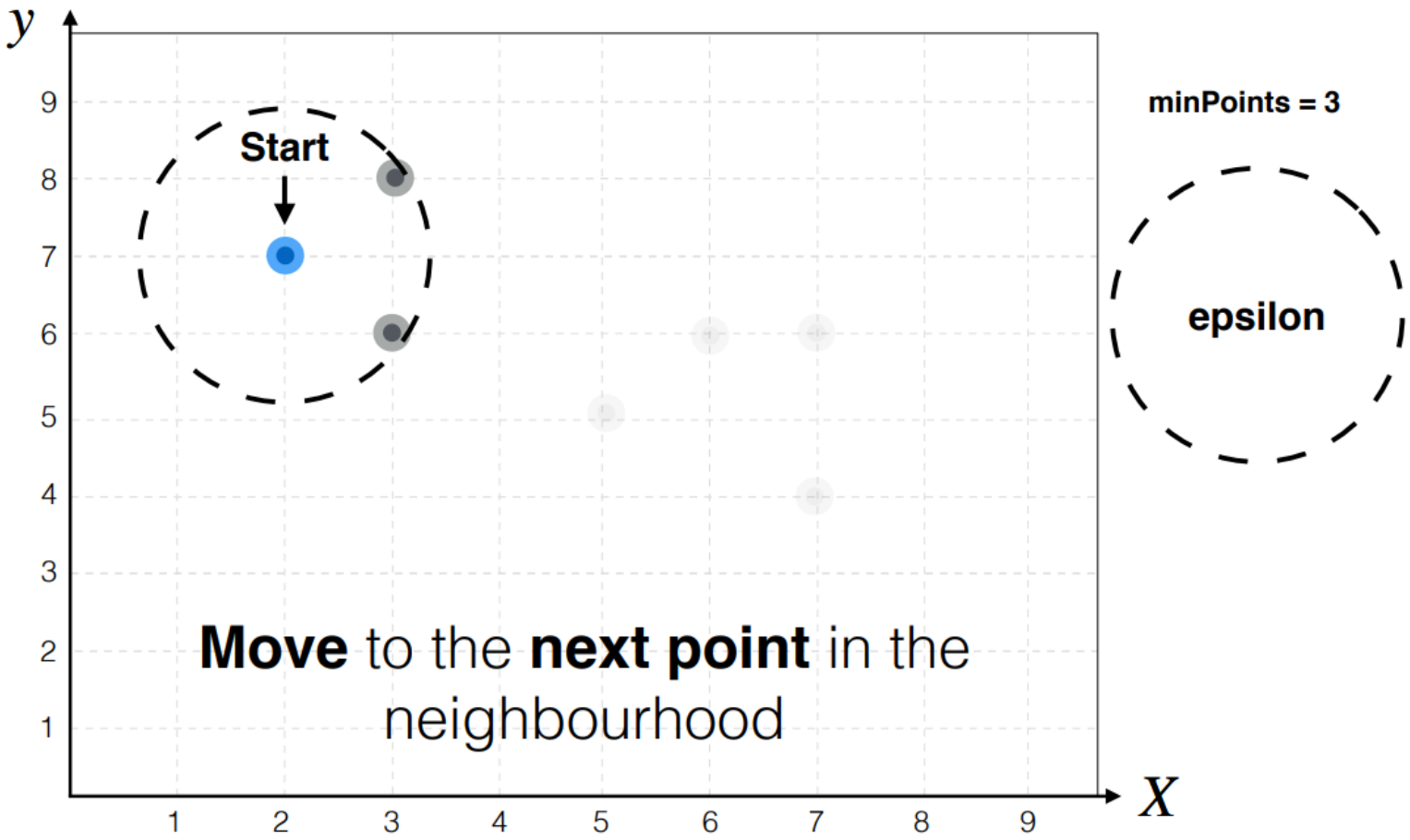
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



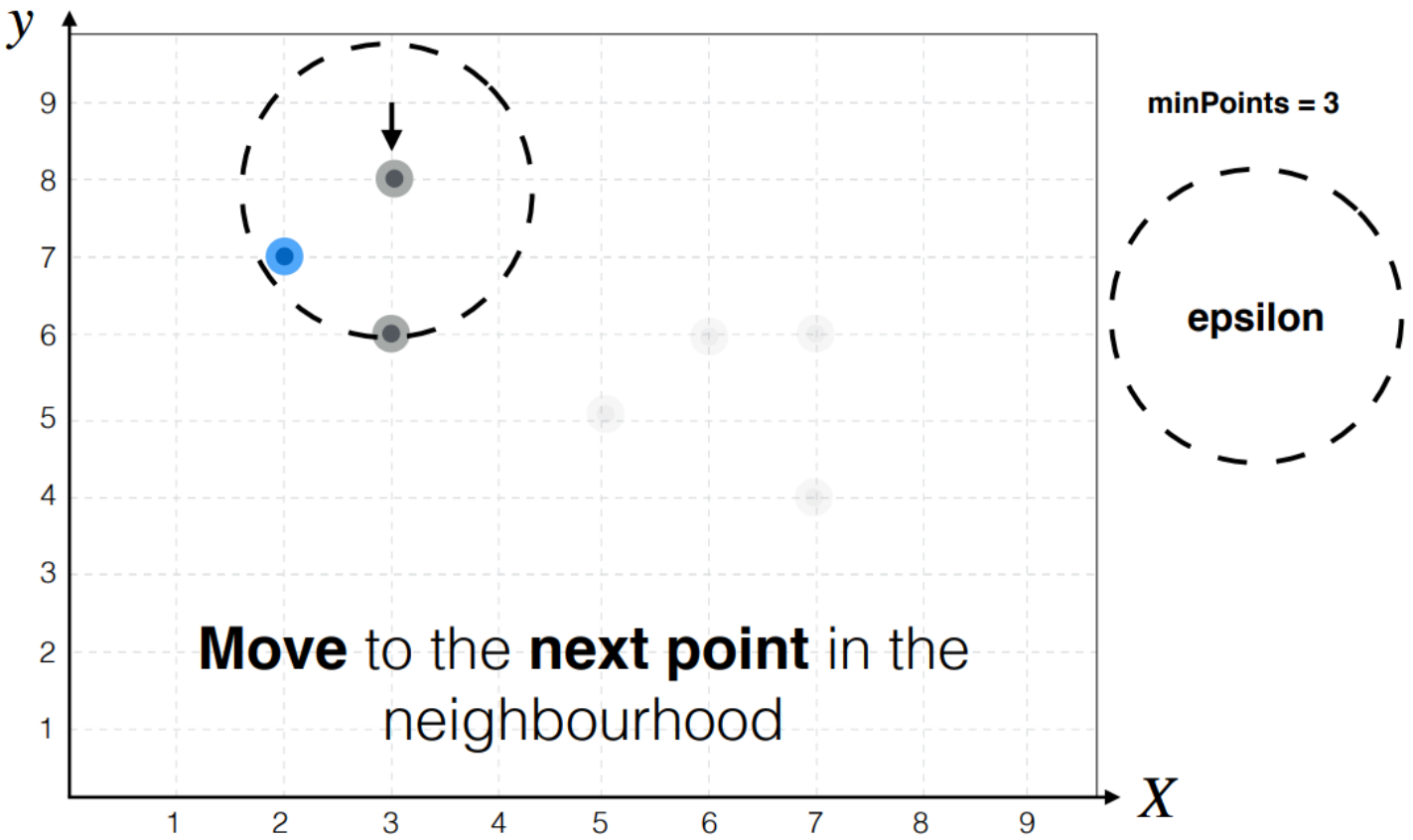
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



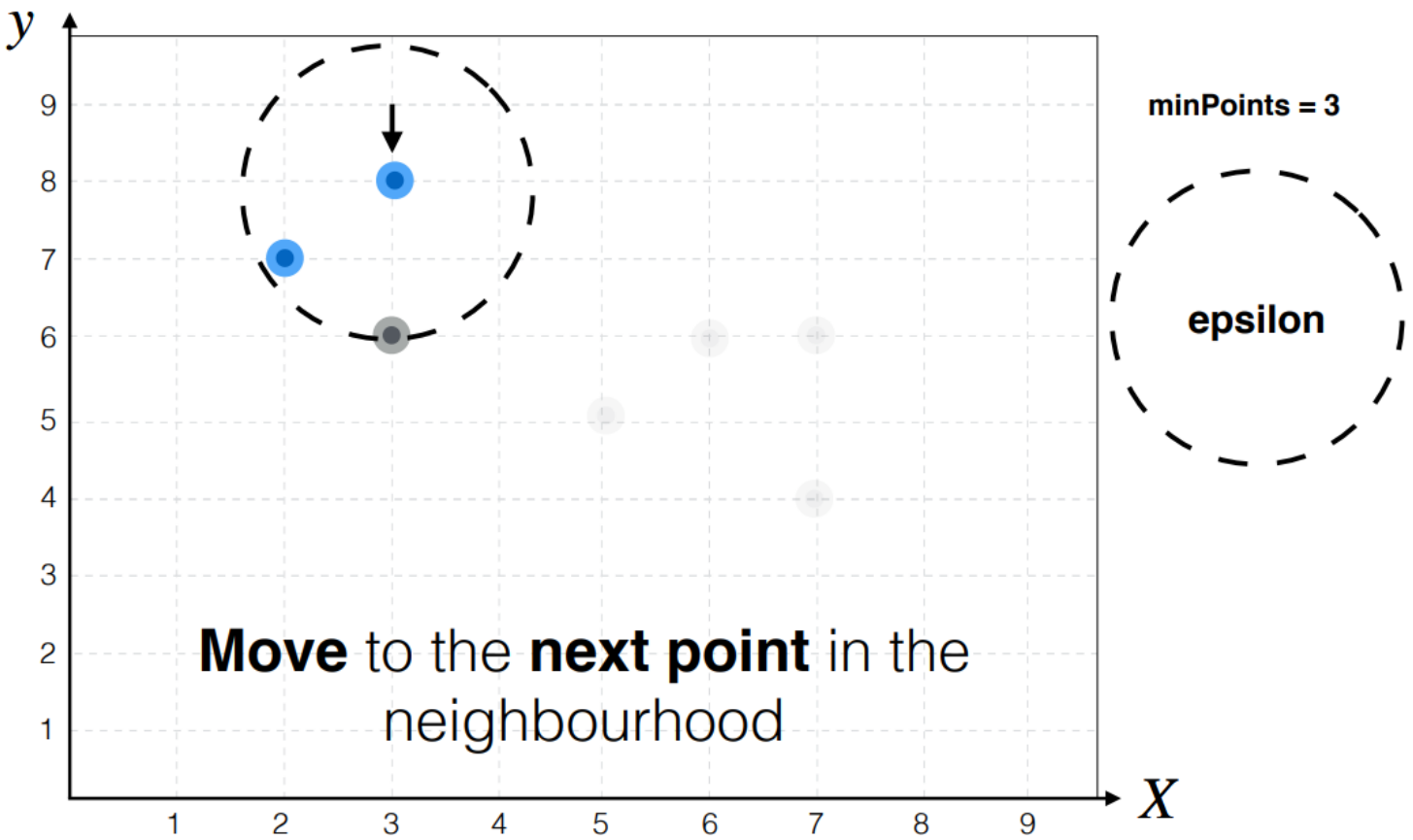
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



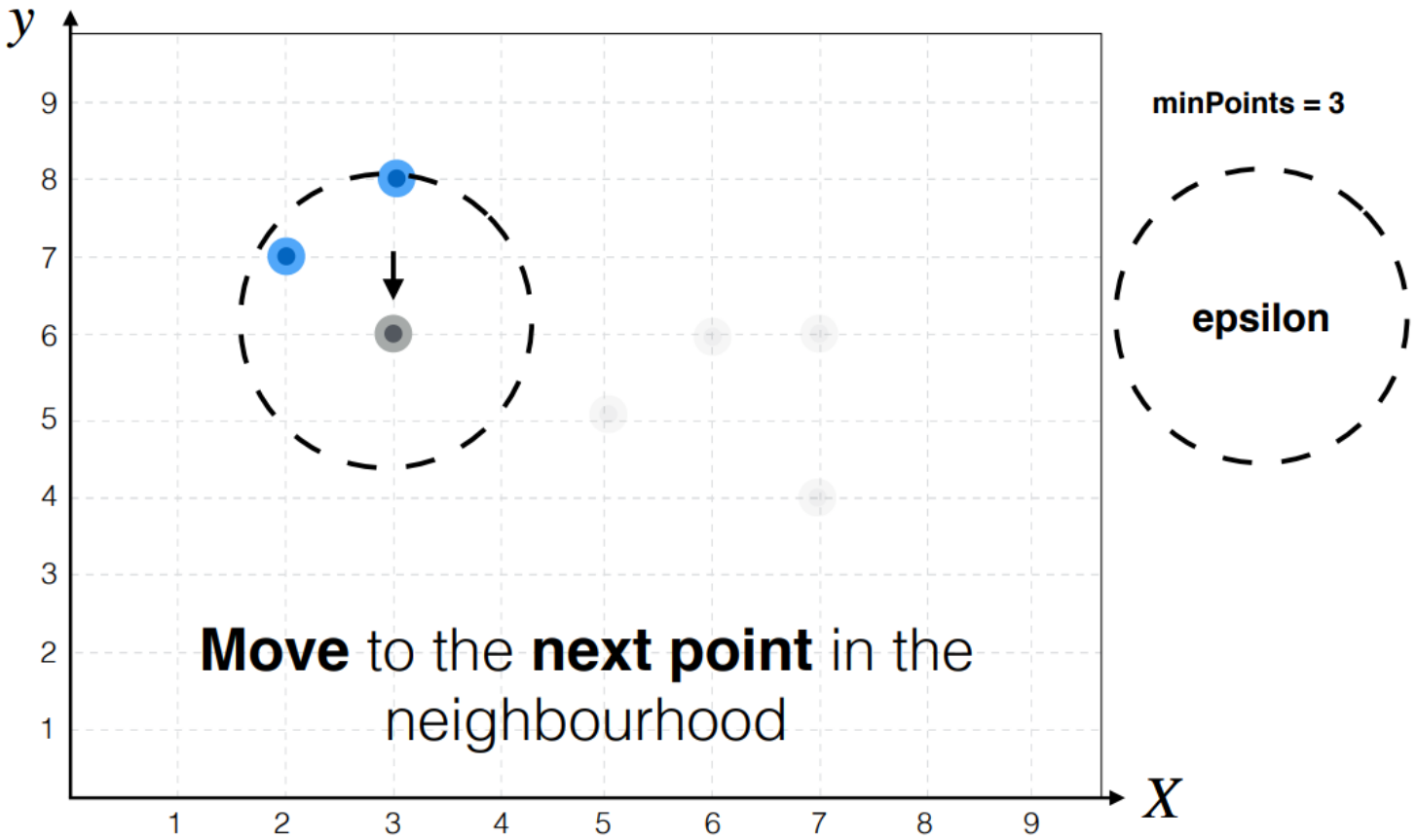
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



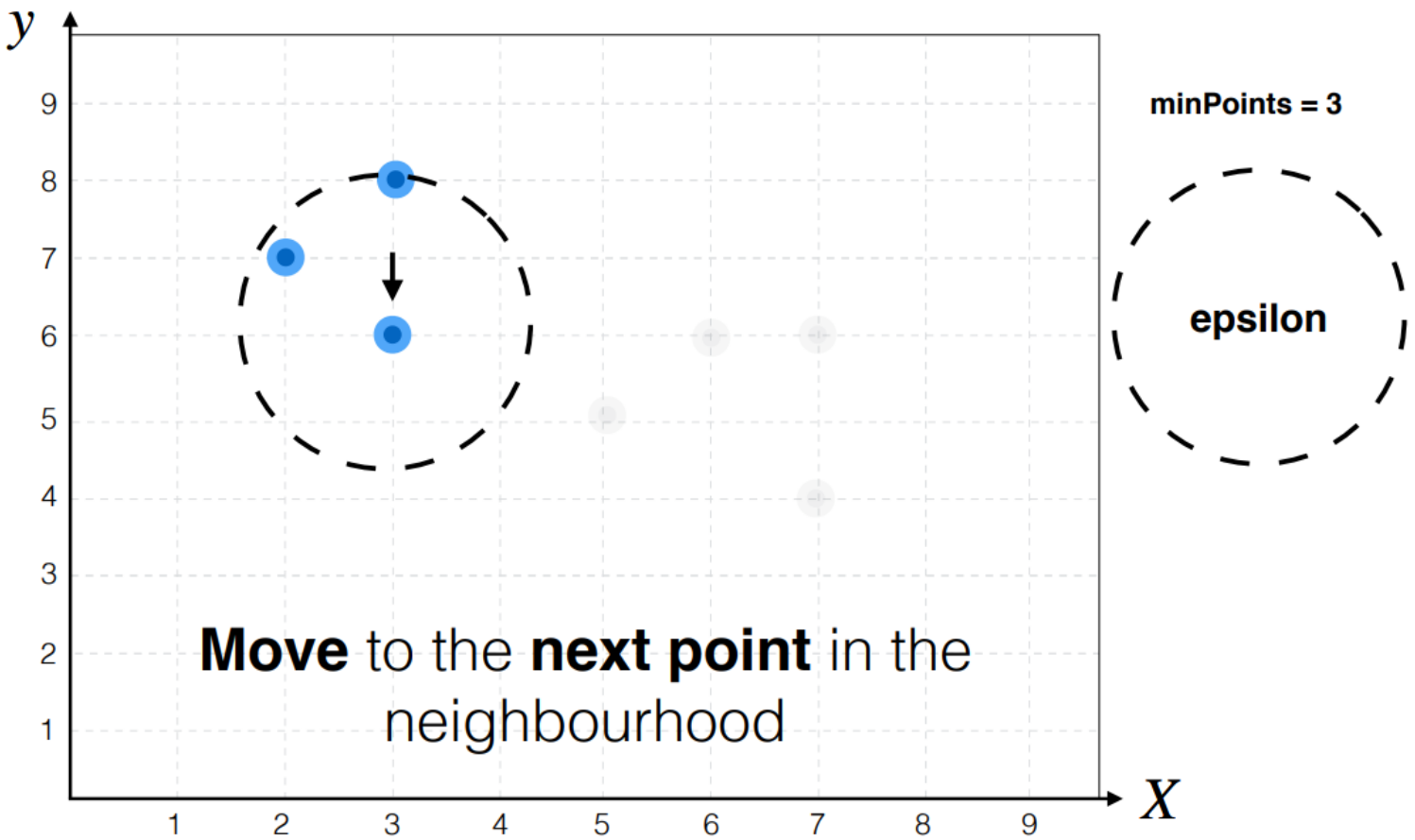
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



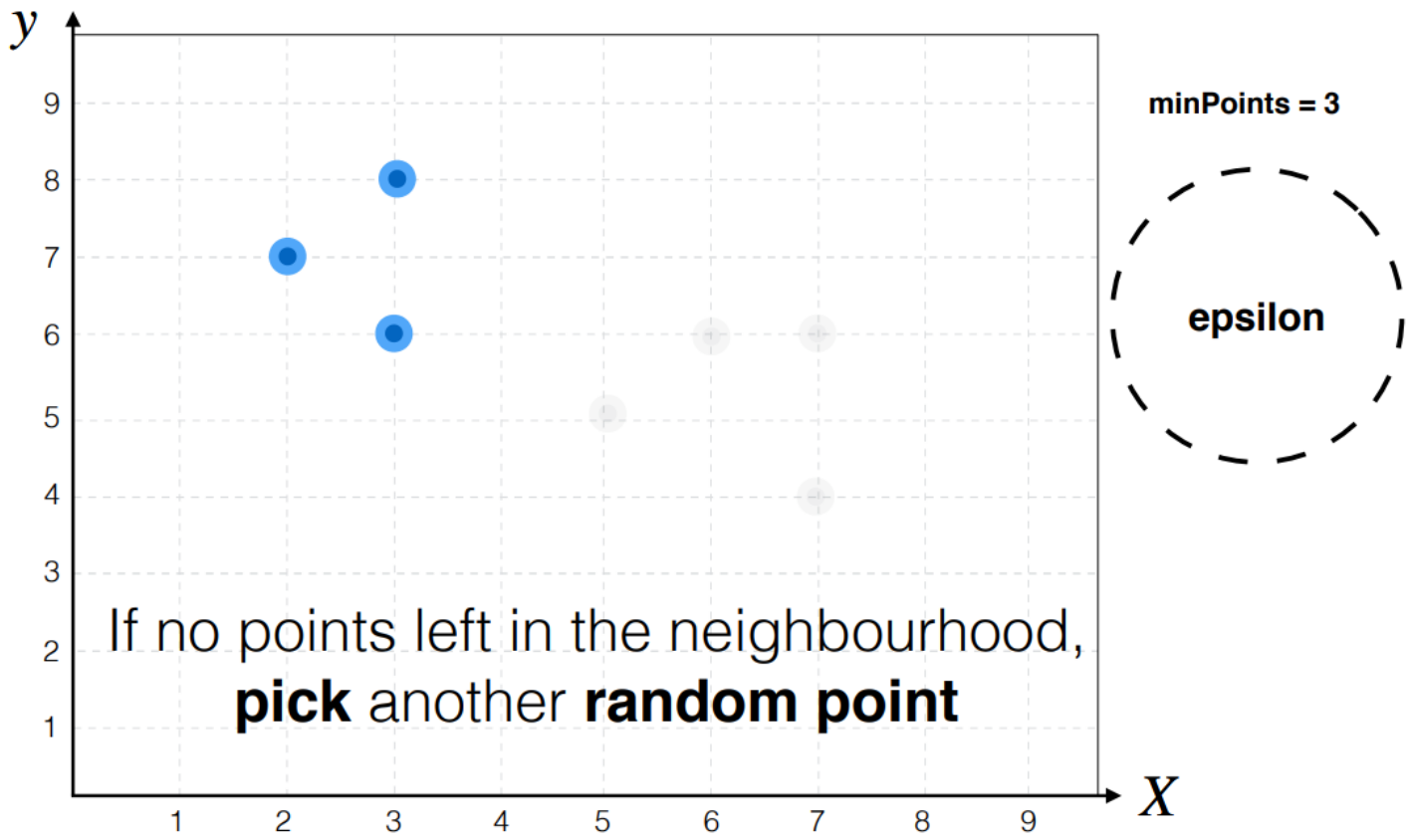
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



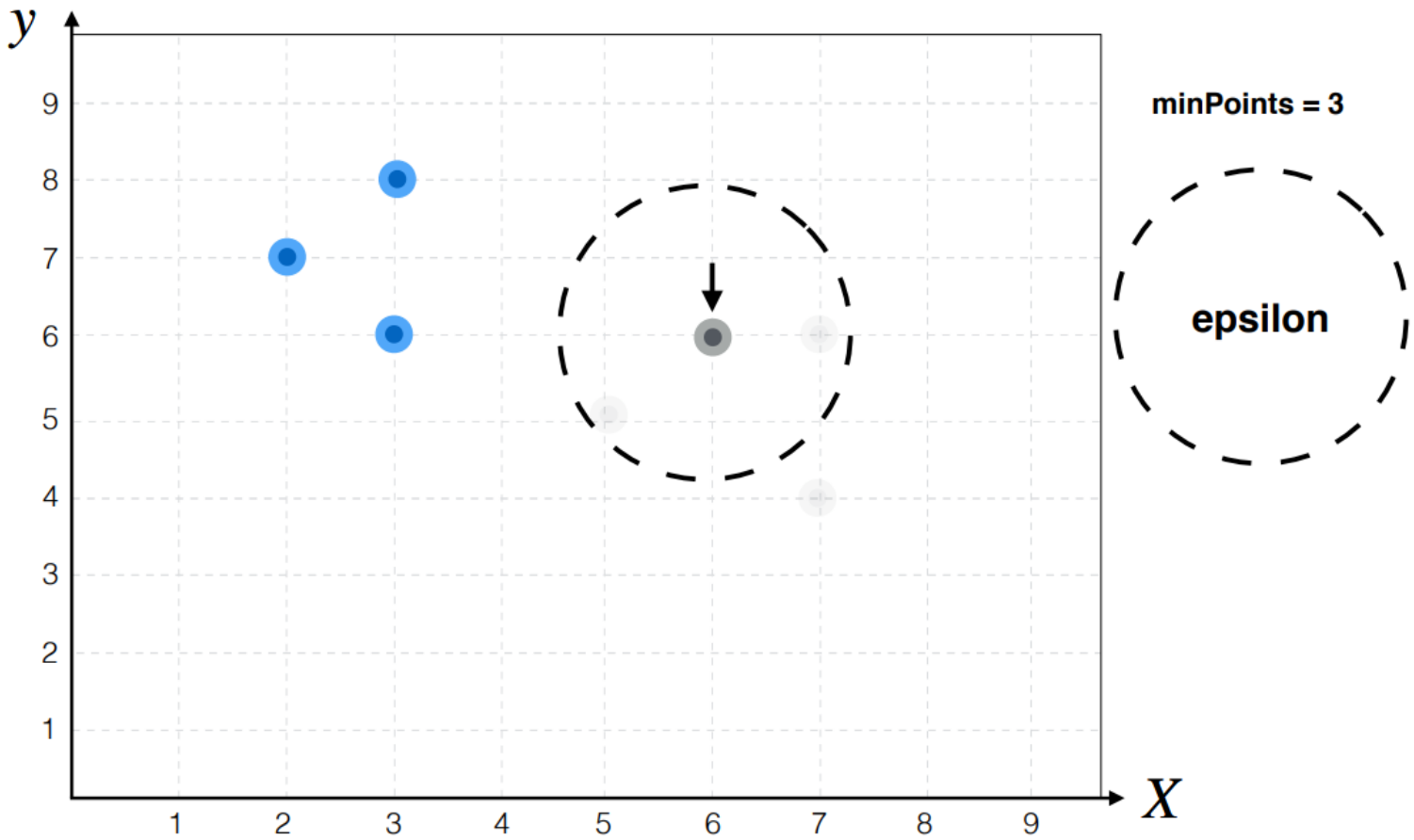
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



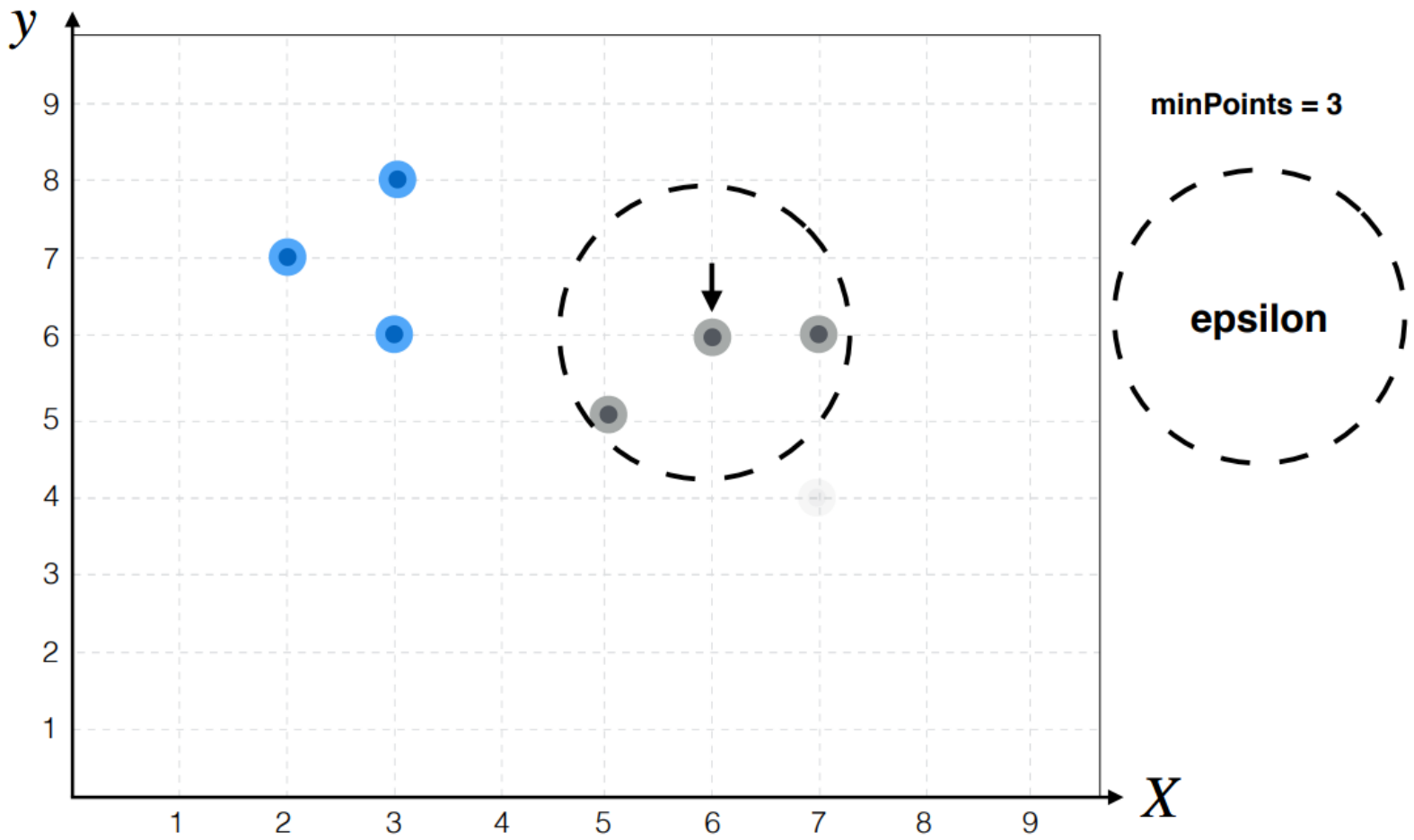
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



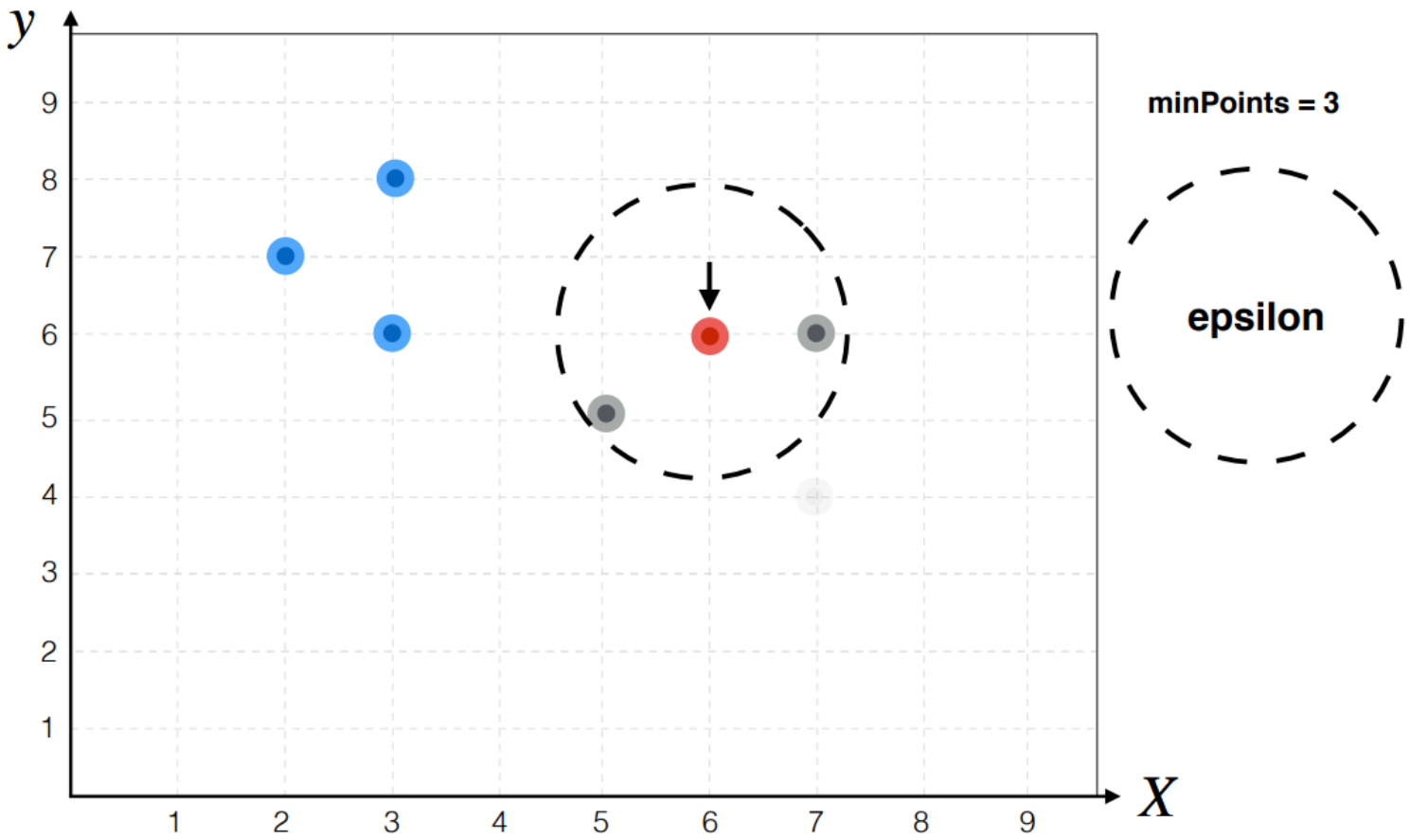
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



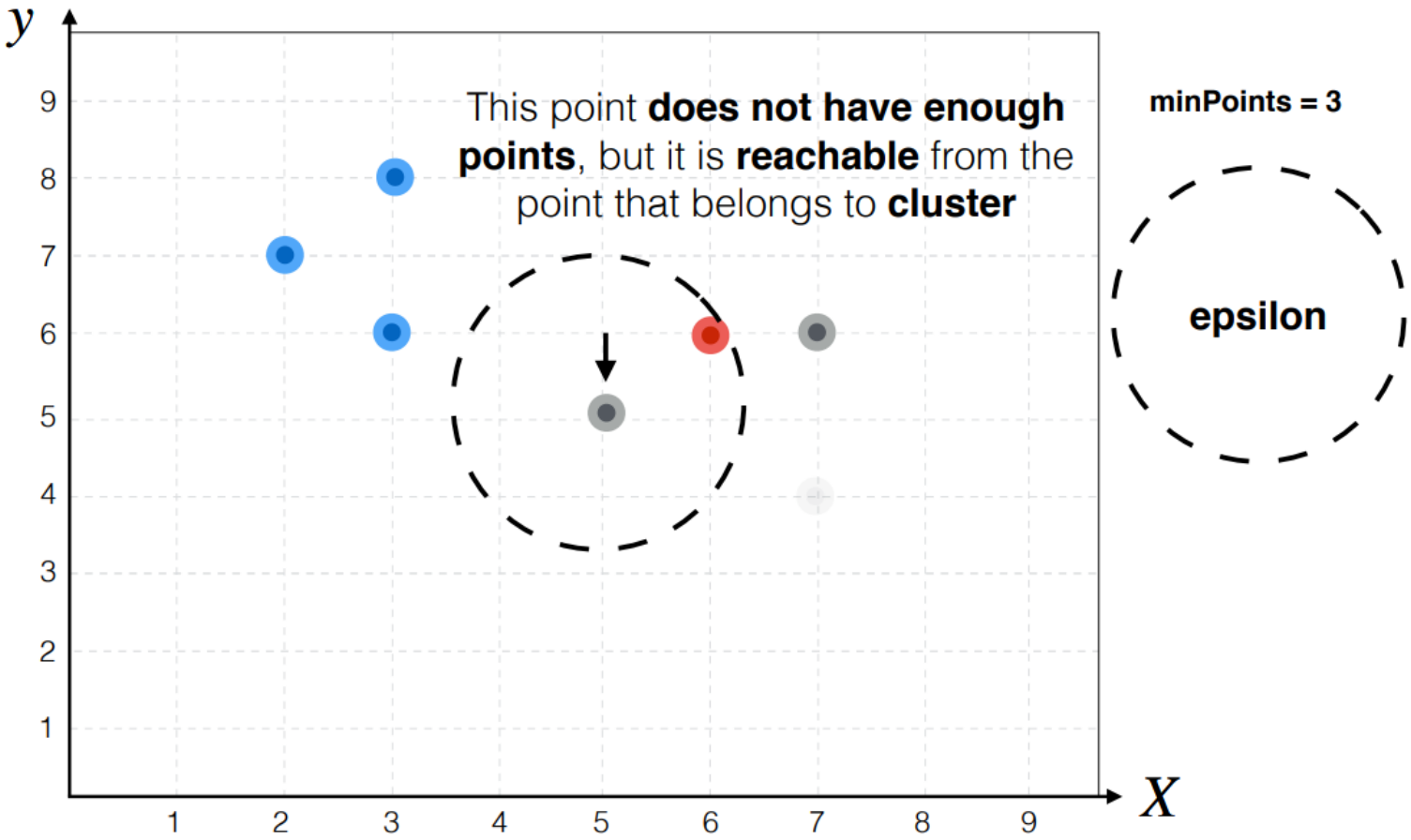
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



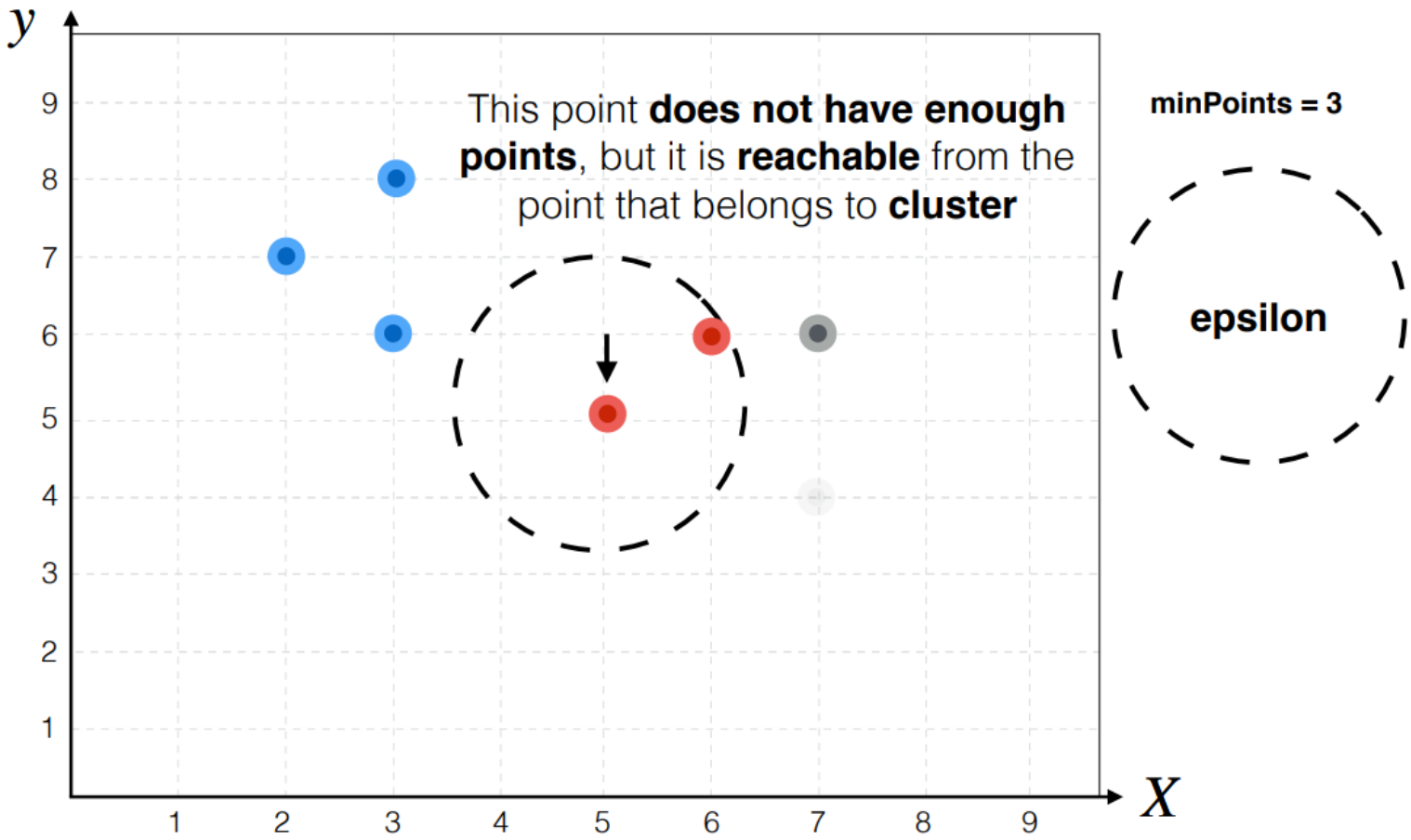
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



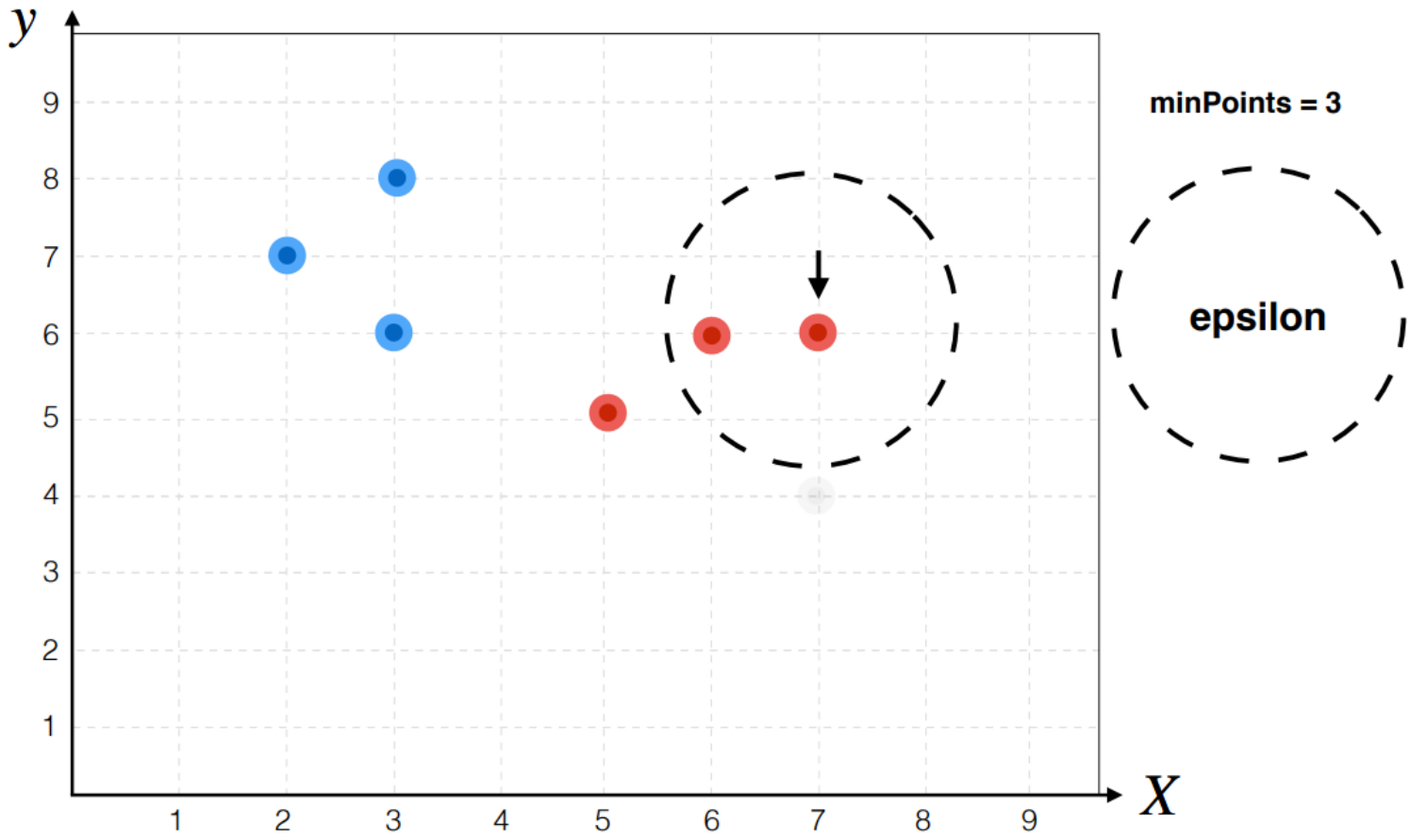
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



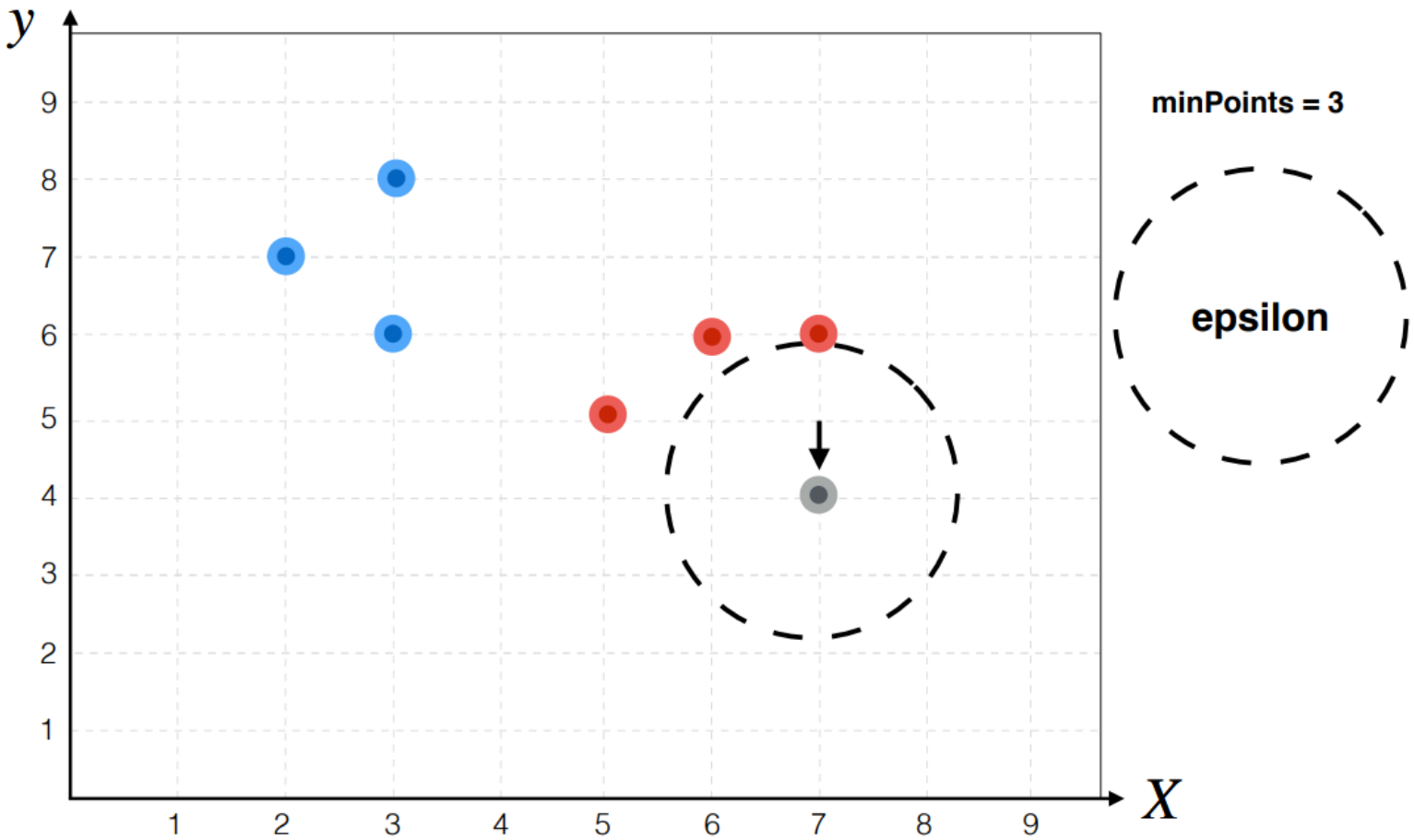
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



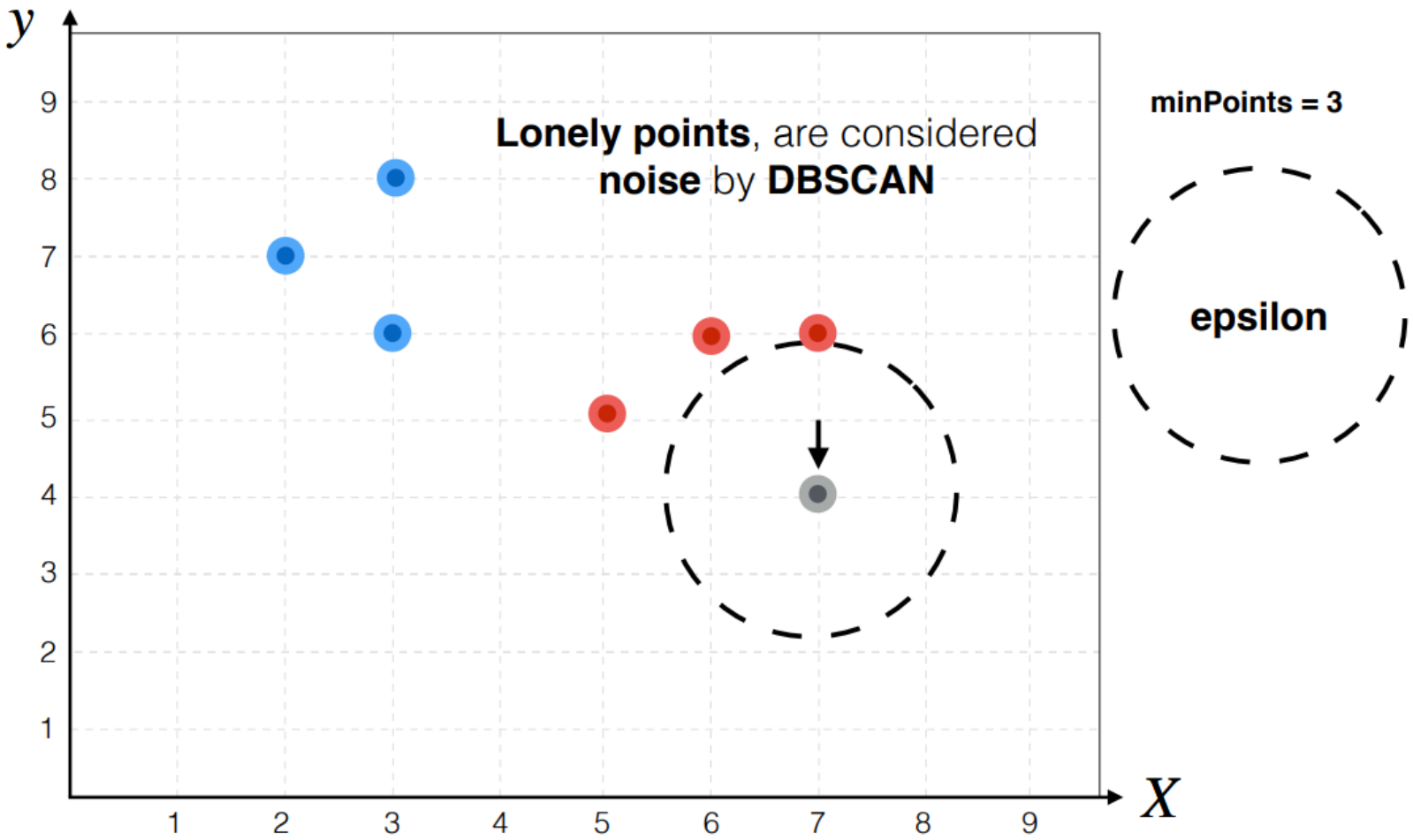
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



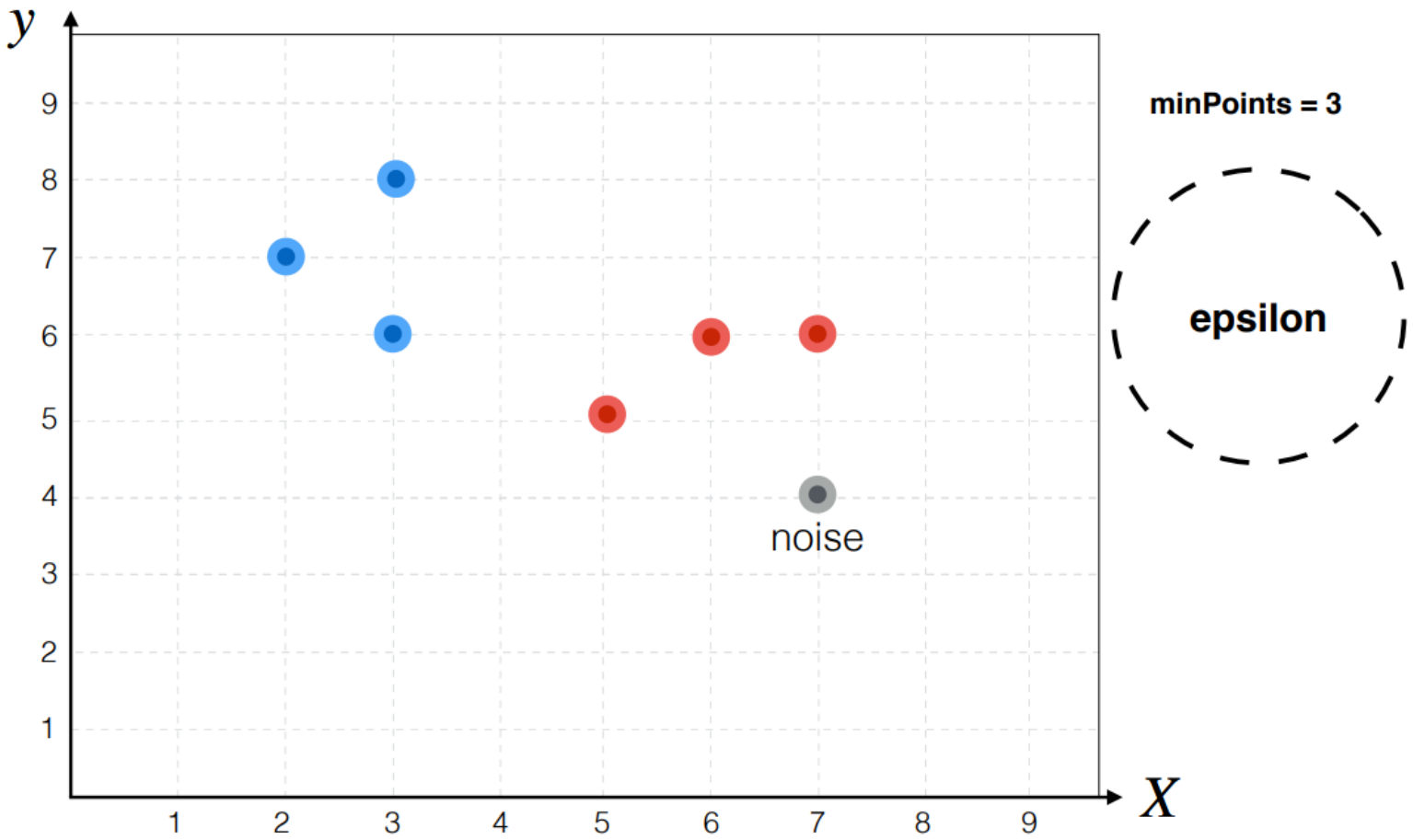
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)



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



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



Distance (dissimilarity) measures

- A key component in cluster analysis is the notion of similarity (or dissimilarity) between the individual data objects being clustered

Distance (dissimilarity) measures

- A key component in cluster analysis is the notion of similarity (or dissimilarity) between the individual data objects being clustered
- How can we measure distances (dissimilarities)?

Distance (dissimilarity) measures

- A key component in cluster analysis is the notion of similarity (or dissimilarity) between the individual data objects being clustered
- How can we measure distances (dissimilarities)?
- Choosing a good distance measure is critically important for clustering

Distance (dissimilarity) measures

- A key component in cluster analysis is the notion of similarity (or dissimilarity) between the individual data objects being clustered
- How can we measure distances (dissimilarities)?
- Choosing a good distance measure is critically important for clustering
- Distance measure formalizes how to describe which objects are "close" to each other, and which are not

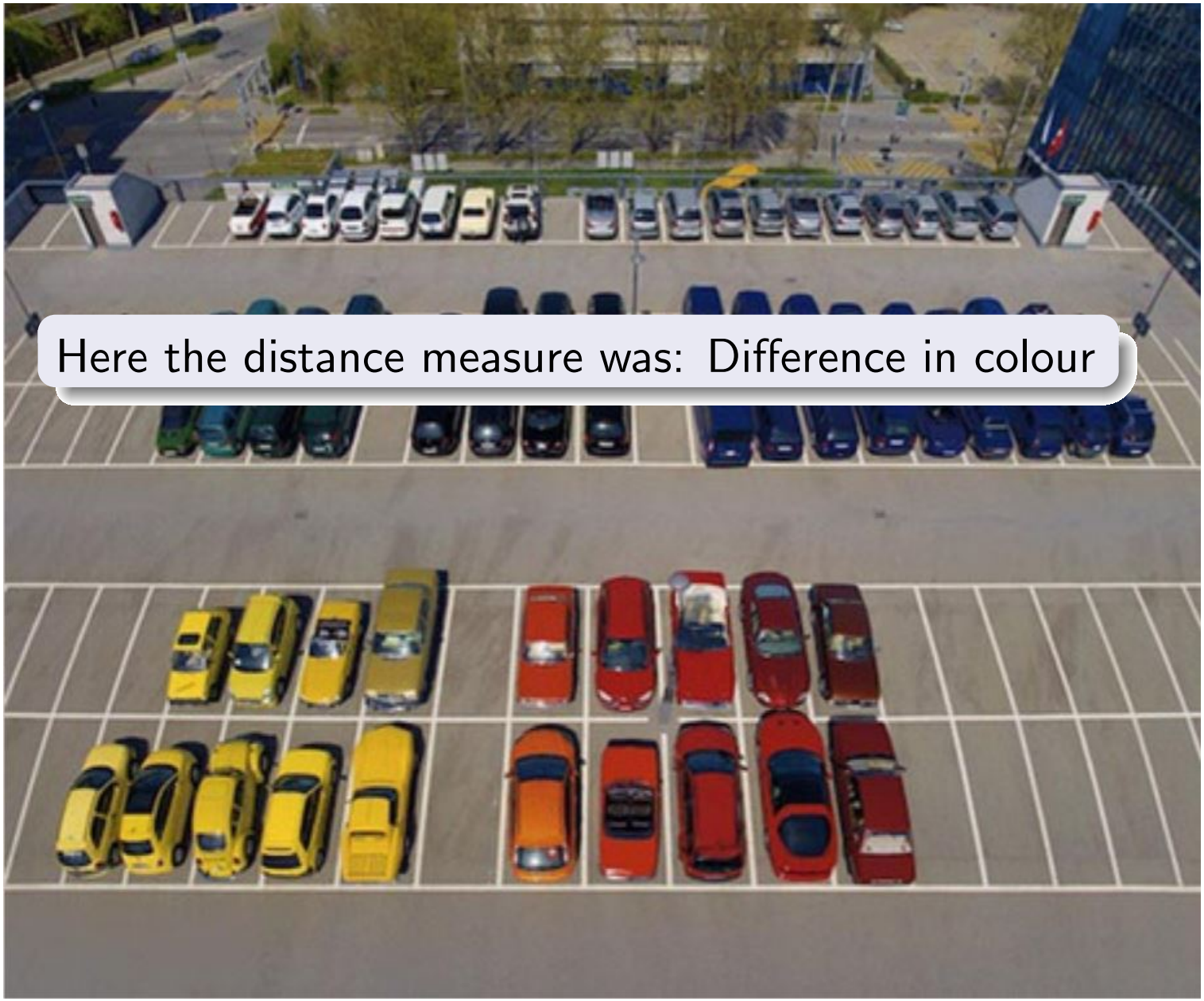
Distance (dissimilarity) measures

- A key component in cluster analysis is the notion of similarity (or dissimilarity) between the individual data objects being clustered
- How can we measure distances (dissimilarities)?
- Choosing a good distance measure is critically important for clustering
- Distance measure formalizes how to describe which objects are "close" to each other, and which are not
- On the same dataset the clustering result can be very different if changing the distance measure

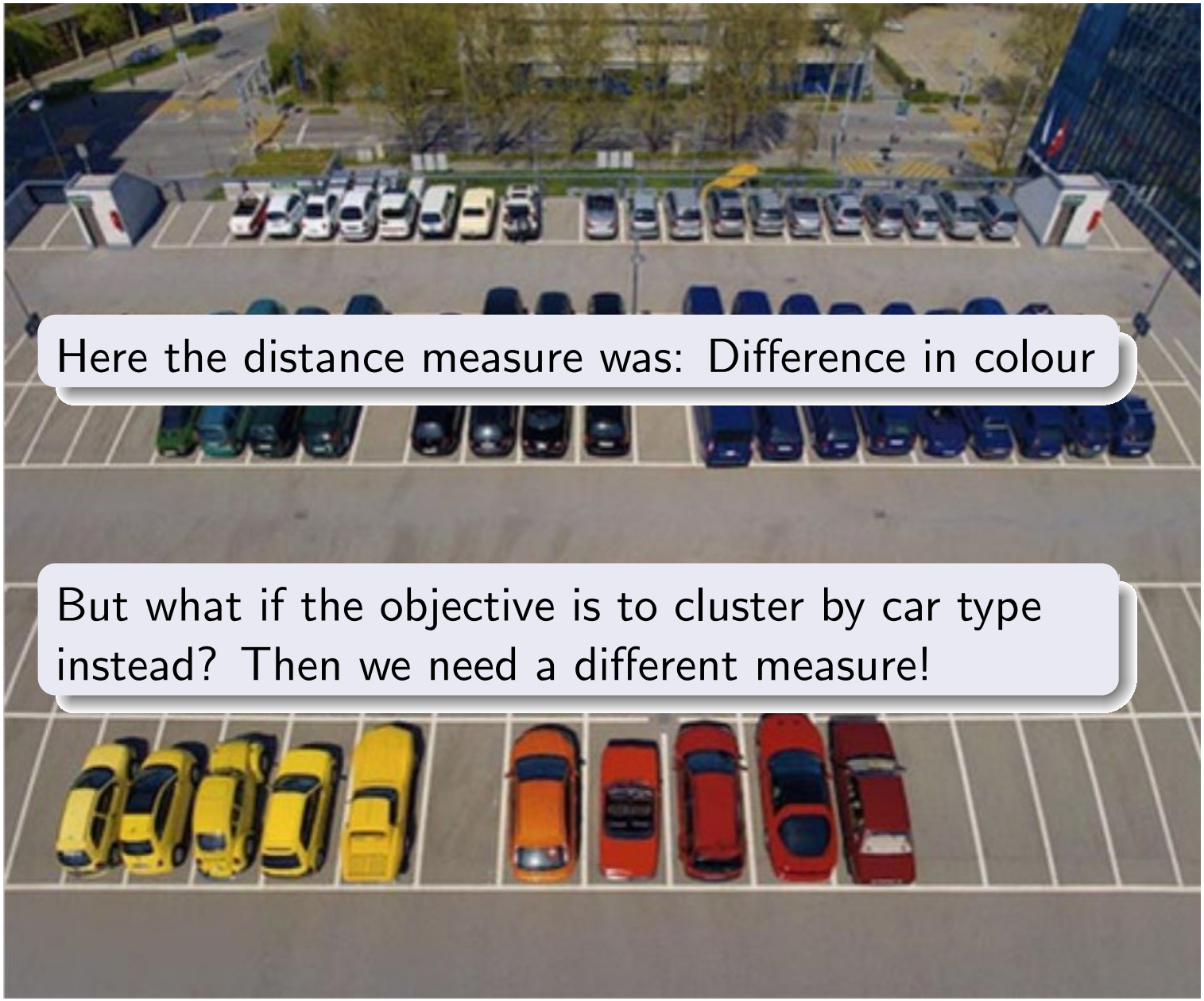
Example: Parking Lot

Example: Parking Lot

Example: Parking Lot



Example: Parking Lot



Distance measure requirements

1. Distance measure is usually required to be a metric:

Distance measure requirements

1. Distance measure is usually required to be a metric:
   - $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$ (positive definite)

Distance measure requirements

1. Distance measure is usually required to be a metric:
   - $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$ (positive definite)
   - $d(x, y) = d(y, x)$ (symmetry)

Distance measure requirements

1. Distance measure is usually required to be a metric:
   - $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$ (positive definite)
   - $d(x, y) = d(y, x)$ (symmetry)
   - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

Distance measure requirements

1. Distance measure is usually required to be a metric:
   - $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$ (positive definite)
   - $d(x, y) = d(y, x)$ (symmetry)
   - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)
2. Sometimes triangle inequality is dropped (semi-metric) and we refer to the measure as **dissimilarity** measure

Choosing dissimilarity measures

1. Decide which features should be included into distance calculation

Choosing dissimilarity measures

1. Decide which features should be included into distance calculation
2. Make all features into numeric by encoding each categorical feature as multiple binary features (one-hot encoding)

Choosing dissimilarity measures

1. Decide which features should be included into distance calculation
2. Make all features into numeric by encoding each categorical feature as multiple binary features (one-hot encoding)
3. Rescale all features so that the difference of size 1 means roughly the same in each feature

Choosing dissimilarity measures

1. Decide which features should be included into distance calculation
2. Make all features into numeric by encoding each categorical feature as multiple binary features (one-hot encoding)
3. Rescale all features so that the difference of size 1 means roughly the same in each feature
4. Think about some well-known measures (euclidean, manhattan etc.), see which makes more sense to use or design your own dissimilarity measure

What have we learned today?

- ✓ Concepts of Cluster Analysis
- ✓ Hierarchical Clustering
- ✓ K-means Clustering
- ✓ K-medoids Clustering
- ✓ DBSCAN
- ✓ Dissimilariy Measure Selection