3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

Clustering
&
Nearest Neighbor Classification

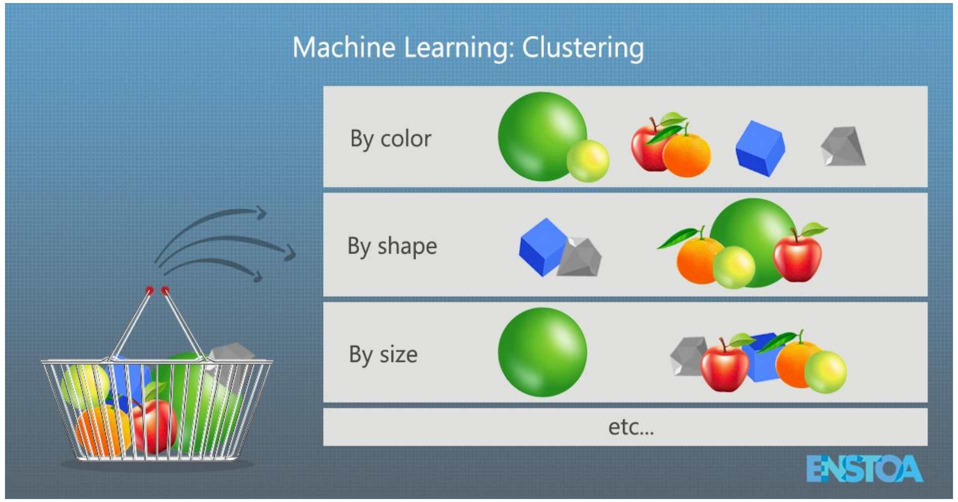
Liangliang Nan
https://3d.bk.tudelft.nl/liangliang/

Agenda

- Overview
  - What is clustering?
  - Distance measure (similarity measure)
  - Types of clustering algorithms
- Clustering algorithms
  - K-means clustering
  - Hierarchical clustering
  - Density-based clustering
- Nearest neighbor classification
- Features

What is clustering?

- Clustering
  - A process that **partitions** a given dataset into homogeneous groups based on given features such that **similar** objects are kept in a group whereas **dissimilar** objects are in different groups.



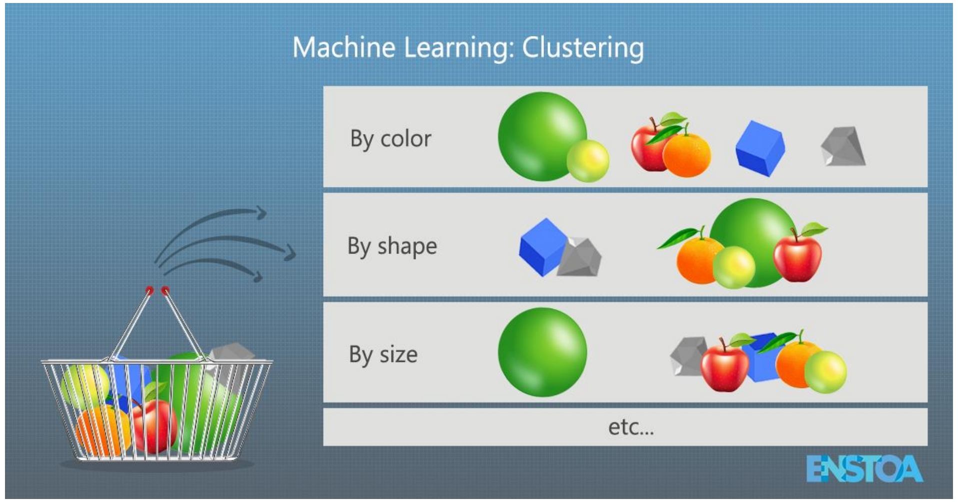
What is a cluster?

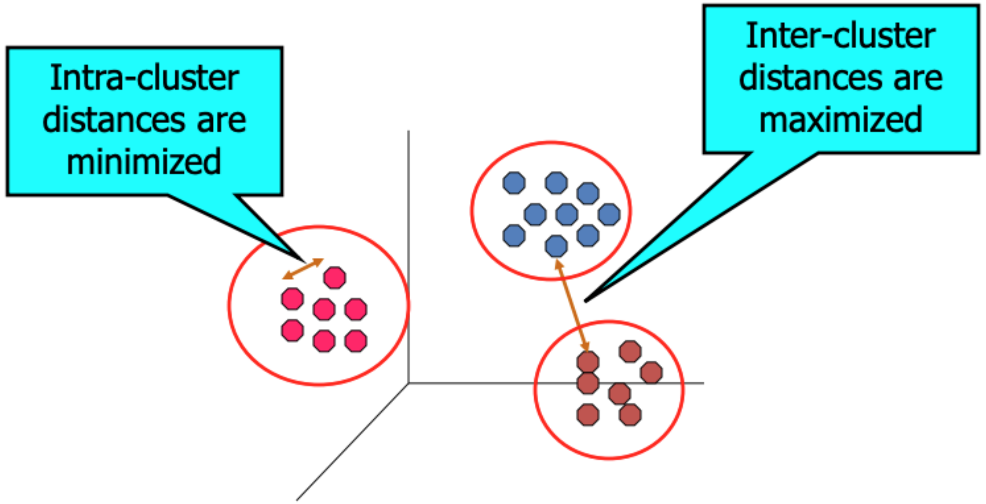
What constitutes a good cluster?

What is the “best” criterion for clustering?

What is clustering?

- Clustering
  - A process that **partitions** a given dataset into homogeneous groups based on given features such that **similar** objects are kept in a group whereas **dissimilar** objects are in different groups.





What is clustering?

- Clustering: two components in an algorithm
  - Distance measure → defines similarities
  - Clustering algorithm → partitions the dataset

Different distance measures lead to different clustering results

Distance measure

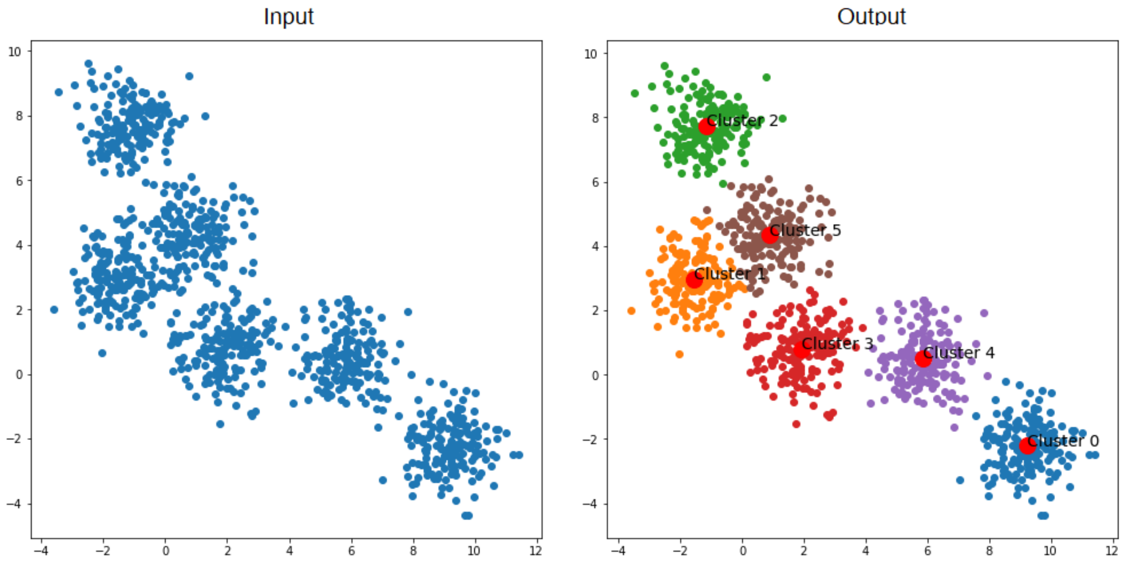
- Problem dependent
  - Minkowski distance/metric is often used
    - Generalization of Euclidean distance (L^2) and Manhattan distance (L^1)

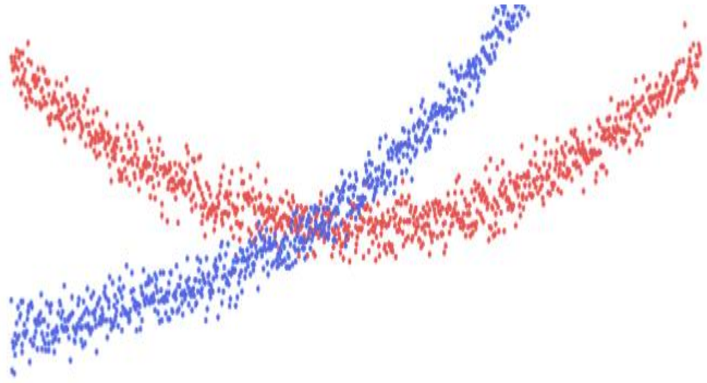
$$D\left(x_i, x_j\right) = \left(\sum_{k=1}^{d} \left|x_{i,k} - x_{j,k}\right|^p\right)^{\frac{1}{p}}$$

  - Domain knowledge is required
    - When components of data feature vectors not immediately comparable, e.g.,
      - color vs size
      - distance to city center vs energy label

Types of clustering algorithms

- Different criteria
  - Exclusive vs overlapping
    - Whether a data point can belong to two or more clusters

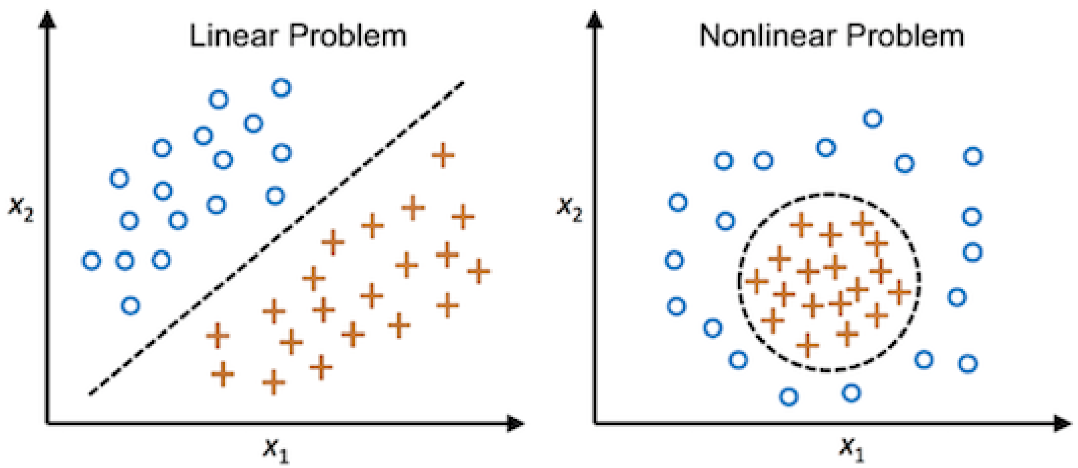


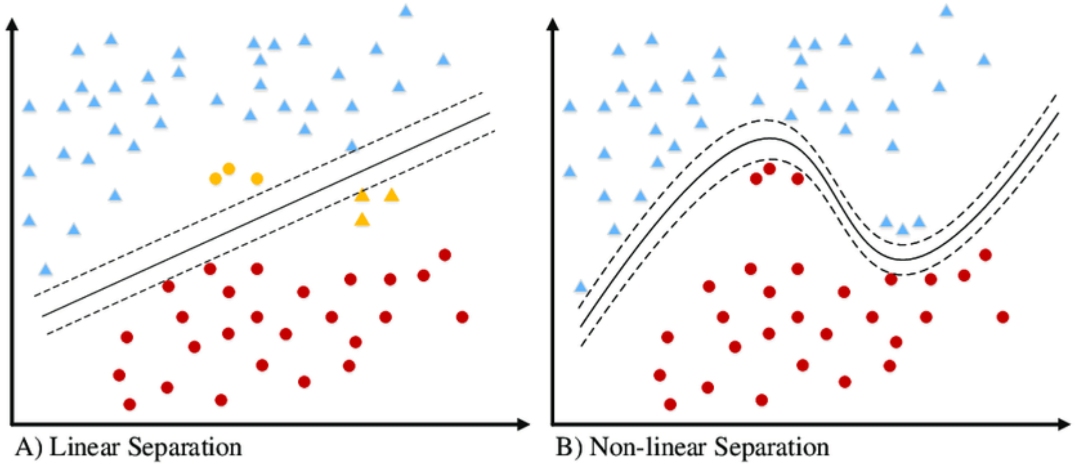


Types of clustering algorithms

- Different criteria
  - Exclusive vs overlapping
    - Whether a data point can belong to two or more clusters
  - Linear vs non-linear
    - The applicability to different types of data

We will learn:
- Linear: K-means, hierarchical clustering
- Non-linear: density-based clustering





Agenda

- Overview
  - What is clustering?
  - Distance measure (similarity measure)
  - Types of clustering algorithms
- Clustering algorithms
  - K-means clustering
  - Hierarchical clustering
  - Density-based clustering
- Nearest neighbor classification
- Features

k-means

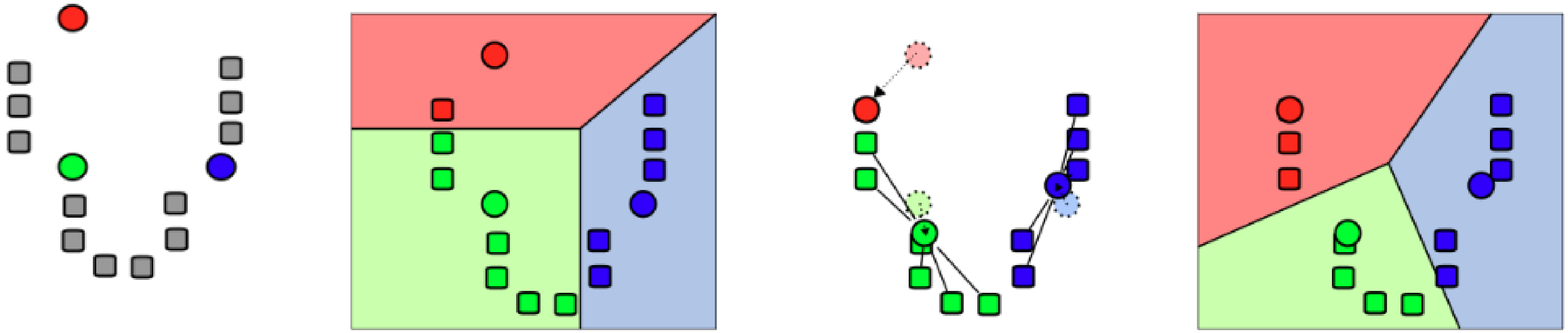
1) Initialize the k clusters $\ell^o = \{c_1^0, c_2^0 \ldots c_k^0\}$ in a way such that the initial centroids are placed as far as possible from each other.

2) Calculate the centroids of the clusters: $u_j^i = \frac{1}{|c_j^i|} \sum_{x \in c_j^i} x$, where $j = 1, ..., k$ and i denotes the i-th iteration.

3) Take each point belonging to a given data set and associate it to the nearest centroid:

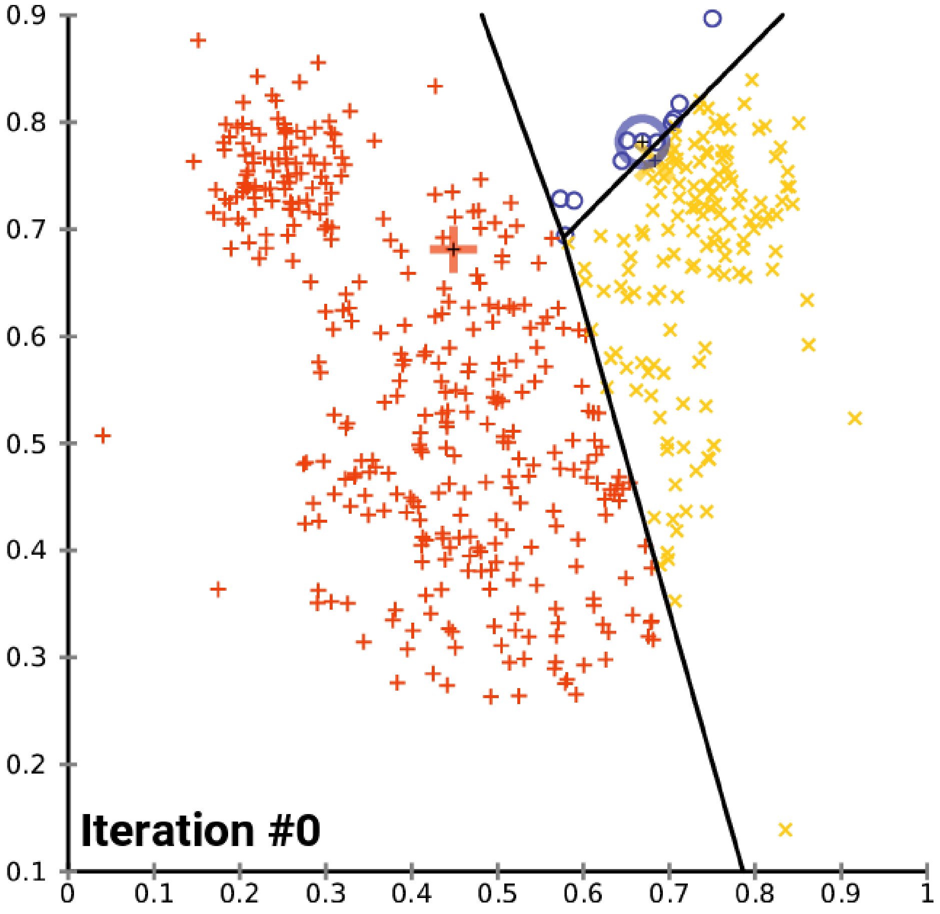
$$\begin{aligned} c_j^{i+1} &= \left\{x \mid d\left(x, u_j^i\right) \leq d\left(x, u_{j'}^i\right), \forall j', 1 \leq j' \leq k\right\} \\ \ell^{i+1} &= \left\{c_j^{i+1} \mid 1 \leq j \leq k\right\} \end{aligned} . \quad (2)$$

4) Repeat steps 2 and 3 until no more changes can be made to the clusters, i.e., $\ell^{i+1} = \ell^i$. In other words, centroids do not move any more.

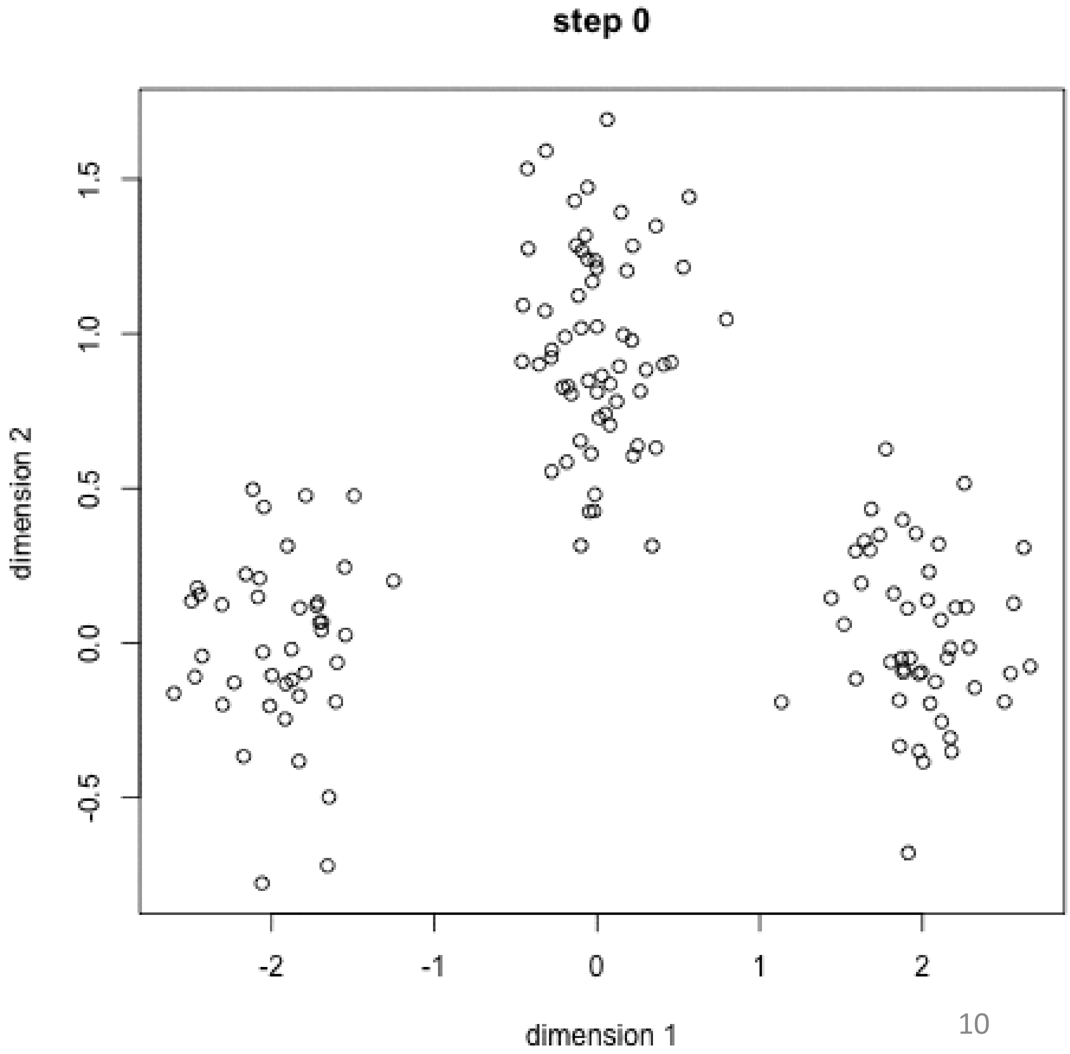


(a) (b) (c) (d)

k-means



Iteration #0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



step 0
dimension 2
dimension 1
1.5
1.0
0.5
0.0
-0.5
-2
-1
0
1
2

k-means: an optimization perspective

- Objective function
  - SSE (Sum of Squared Error) $J = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - u_i\|^2$

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

k-means: an optimization perspective

- Objective function
  - SSE (Sum of Squared Error) $J = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - u_i\|^2$
- Convergence
  - K-means is exactly coordinate descent on *J*
    - Step 2: fixed cluster assignment—compute cluster centroids that minimize the current error
    - Step 3: fixed cluster centroids—find cluster assignment that minimizes the current error

1) Initialize the k clusters $\ell^o = \{c_1^0, c_2^0 \ldots c_k^0\}$ in a way such that the initial centroids are placed as far as possible from each other.

2) Calculate the centroids of the clusters: $u_j^i = \frac{1}{|c_j^i|} \sum_{x \in c_j^i} x$, where $j = 1, \ldots, k$ and i denotes the i-th iteration.

3) Take each point belonging to a given data set and associate it to the nearest centroid:

$$\begin{aligned} c_j^{i+1} &= \left\{x \mid d\left(x, u_j^i\right) \leq d\left(x, u_{j'}^i\right), \forall j', 1 \leq j' \leq k\right\} \\ \ell^{i+1} &= \left\{c_j^{i+1} \mid 1 \leq j \leq k\right\} \end{aligned} \quad (2)$$

4) Repeat steps 2 and 3 until no more changes can be made to the clusters, i.e., $\ell^{i+1} = \ell^i$. In other words, centroids do not move any more.

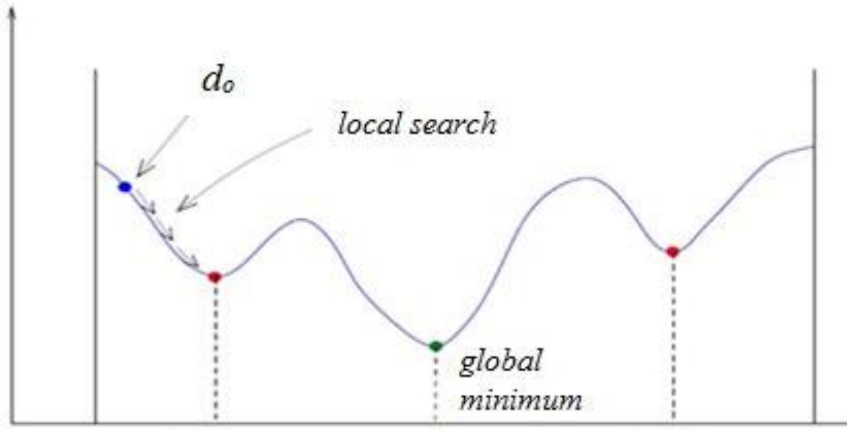
k-means: an optimization perspective

- Objective function
  - SSE (Sum of Squared Error) $J = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - u_i\|^2$
- Convergence
  - K-means is exactly coordinate descent on *J*
    - Step 2: fixed cluster assignment—compute cluster centroids that minimize the current error
    - Step 3: fixed cluster centroids—find cluster assignment that minimizes the current error

J converges a global minimum?

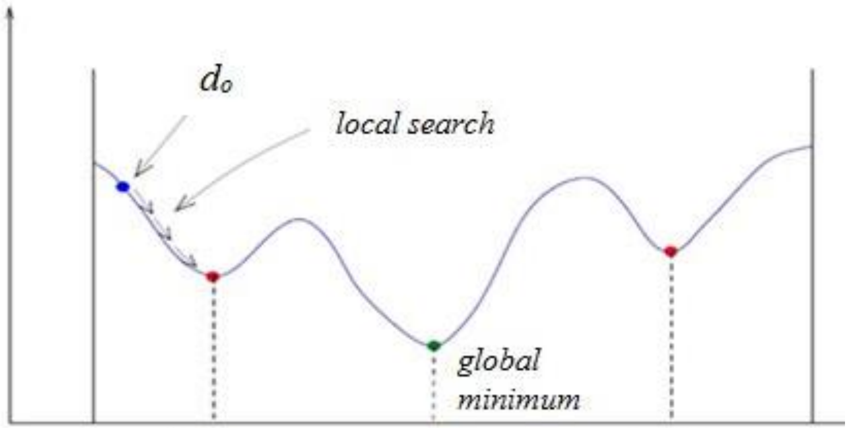
k-means: an optimization perspective

- Objective function
  - SSE (Sum of Squared Error) $J = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - u_i\|^2$
- Convergence
  - K-means is exactly coordinate descent on *J*
    - Step 2: fixed cluster assignment—compute cluster centroids that minimize the current error
    - Step 3: fixed cluster centroids—find cluster assignment that minimizes the current error
- Not necessarily the optimal configuration
  - i.e., local minimum of the objective function



k-means: an optimization perspective

- Objective function
  - SSE (Sum of Squared Error) $$J = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - u_i\|^2$$
- Convergence
  - K-means is exactly coordinate descent on *J*
    - Step 2: fixed cluster assignment—compute cluster centroids that minimize the current error
    - Step 3: fixed cluster centroids—find cluster assignment that minimizes the current error
- Not necessarily the optimal configuration
  - i.e., local minimum of the objective function
  - Solution: repeat many times and pick the best
  - Best configuration not guaranteed



k-means

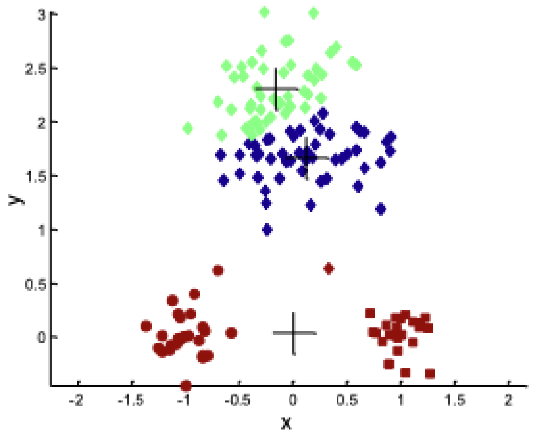
- Advantages
  - Fast and efficient
  - Given good results when groups are distinct or well separated from each other
  - Easy to implement

k-means

- Advantages
  - Fast and efficient
  - Given good results when groups are distinct or well separated from each other
  - Easy to implement
- Limitations

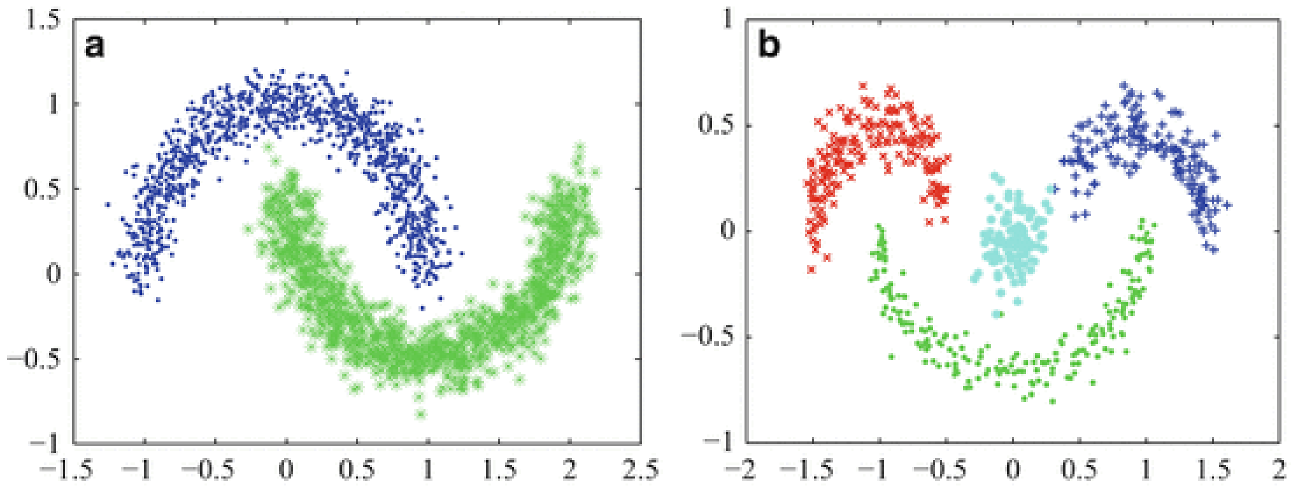
k-means

- Advantages
  - Fast and efficient
  - Given good results when groups are distinct or well separated from each other
  - Easy to implement
- Limitations
  - Requires a priori specification of the number (i.e., k) of clusters
  - Local minima
    - Sensitive to initialization
    - Cannot guarantee optimal clusters
  - Not invariant to non-linear transformations
    - e.g., cartesian coordinates vs polar coordinates



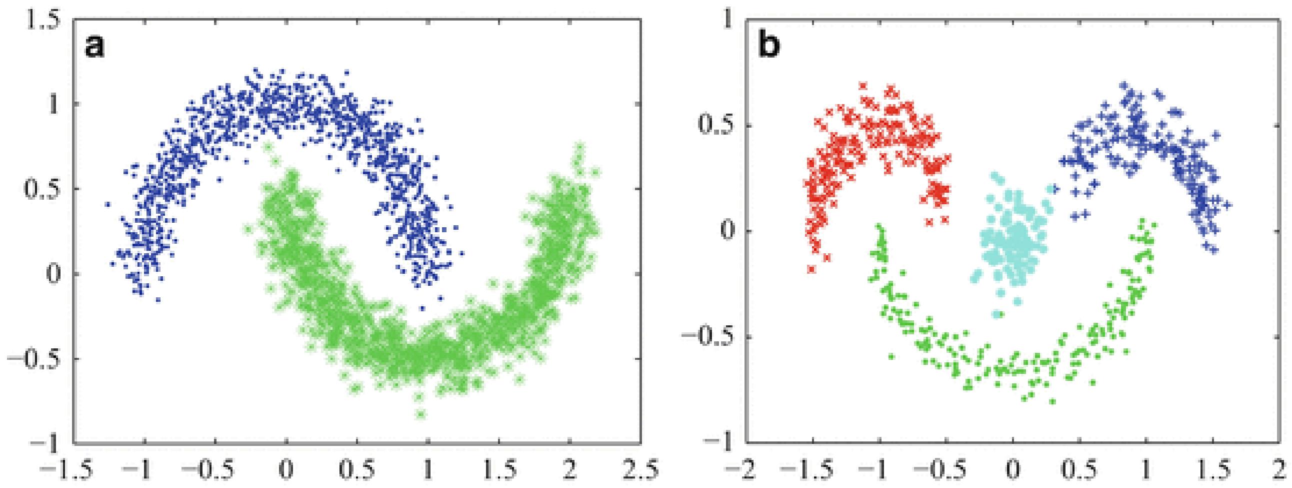
k-means

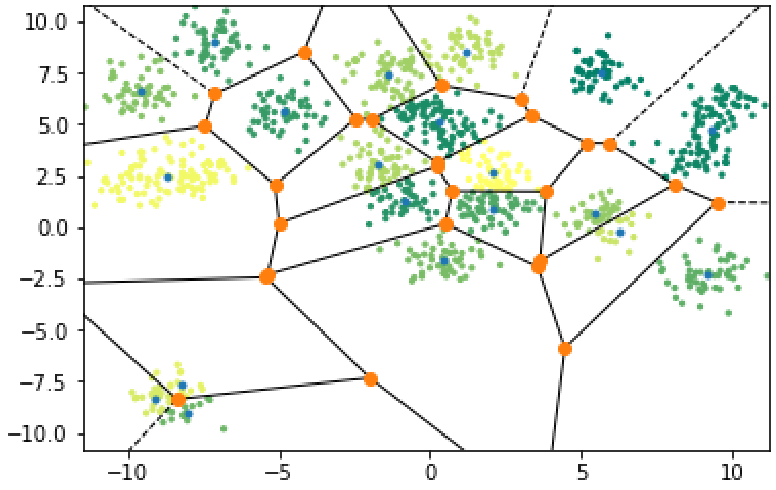
- Can k-means handle?



k-means

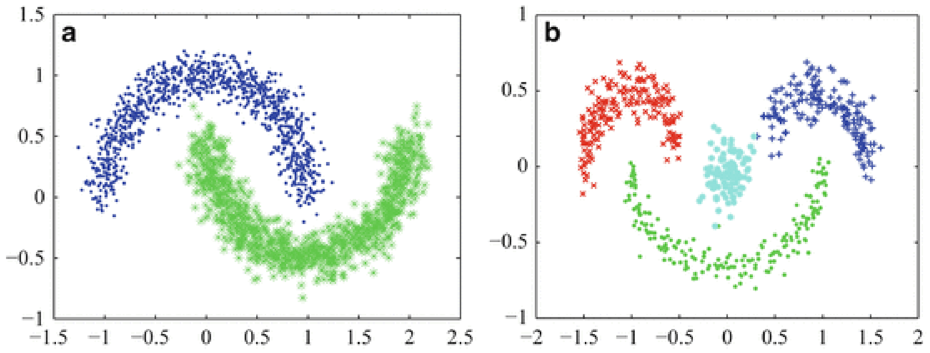
- Can k-means handle?





k-means

- Advantages
  - Fast and efficient
  - Given good results when groups are distinct or well separated from each other
  - Easy to implement
- Limitations
  - Requires a priori specification of the number (i.e., k) of clusters
  - Local minima
    - Sensitive to initialization
    - Cannot guarantee optimal clusters
  - Not invariant to non-linear transformations
    - e.g., cartesian coordinates vs polar coordinates
  - Cannot process non-linear datasets



Agenda

- Overview
  - What is clustering?
  - Distance measure (similarity measure)
  - Types of clustering algorithms
- Clustering algorithms
  - K-means clustering
  - Hierarchical clustering
  - Density-based clustering
- Nearest neighbor classification
- Features

Hierarchical clustering

Given a set of N objects $S = \{s_1, s_2, ..., s_N\}$ to be clustered and a function of distance between two clusters c_i and c_j, build a hierarchy tree on S such that for every $c_i, c_j \in S$, $c_i \cap c_j = \emptyset$. The basic process of hierarchical clustering is as follows:

1) Start by assigning each object to a cluster $c_i = s_i (i = 1, ..., N)$, so that if you have N objects, you have N clusters $\ell = \{c_1, c_2, ..., c_N\}$, each containing just one item.

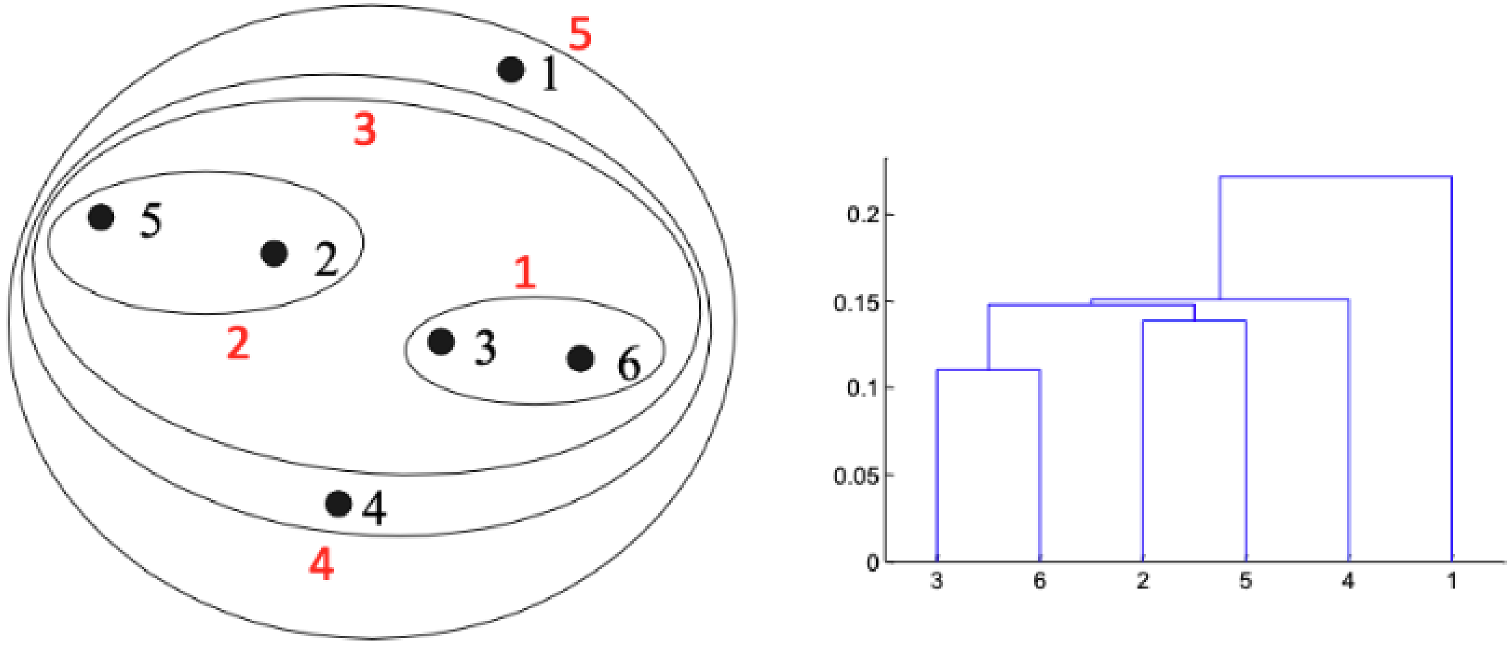
2) Find the pair of clusters (c_i, c_j) such that $D(c_i, c_j) \leq D(c_{i'}, c_{j'})$, $\forall c_{i'} \neq c_{j'} \in \ell$ and merge them into a single cluster $c_k = c_i \cup c_j$. Delete c_i and c_j from ℓ and insert c_k into ℓ so that now you have one cluster less.

3) Compute distances (similarities) between the new cluster and each of the old clusters.

4) Repeat steps 2) and 3) until all items are clustered into a single cluster of size N.

Hierarchical clustering

- Example



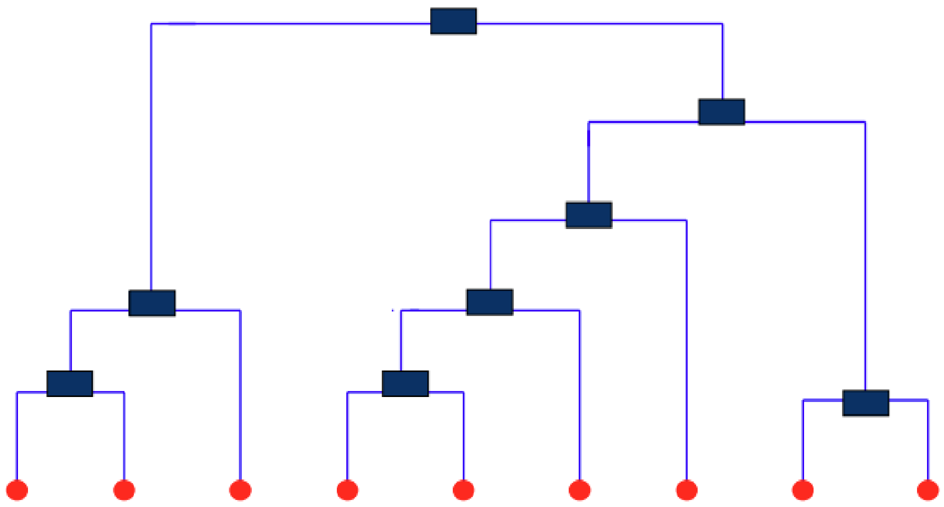
Nested Clusters

Dendrogram

An example of hierarchical clustering

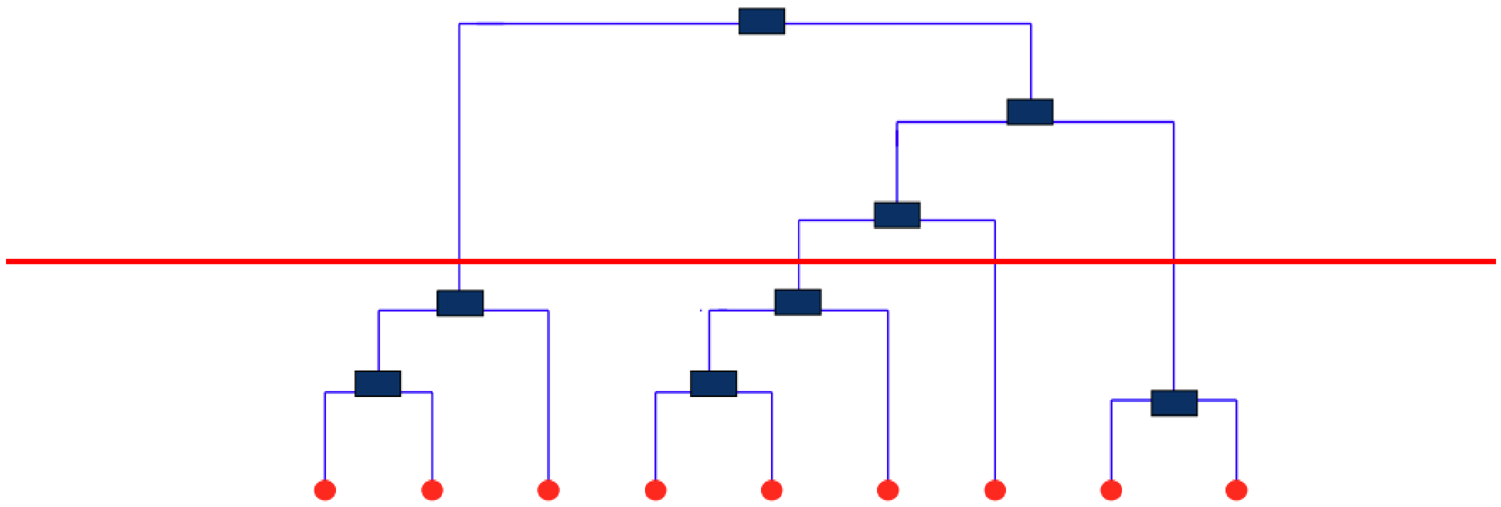
Hierarchical clustering

- Dendrogram
  - A tree that shows how clusters are merged/split hierarchically
  - Each node on the tree is a cluster; each leaf node is a singleton cluster



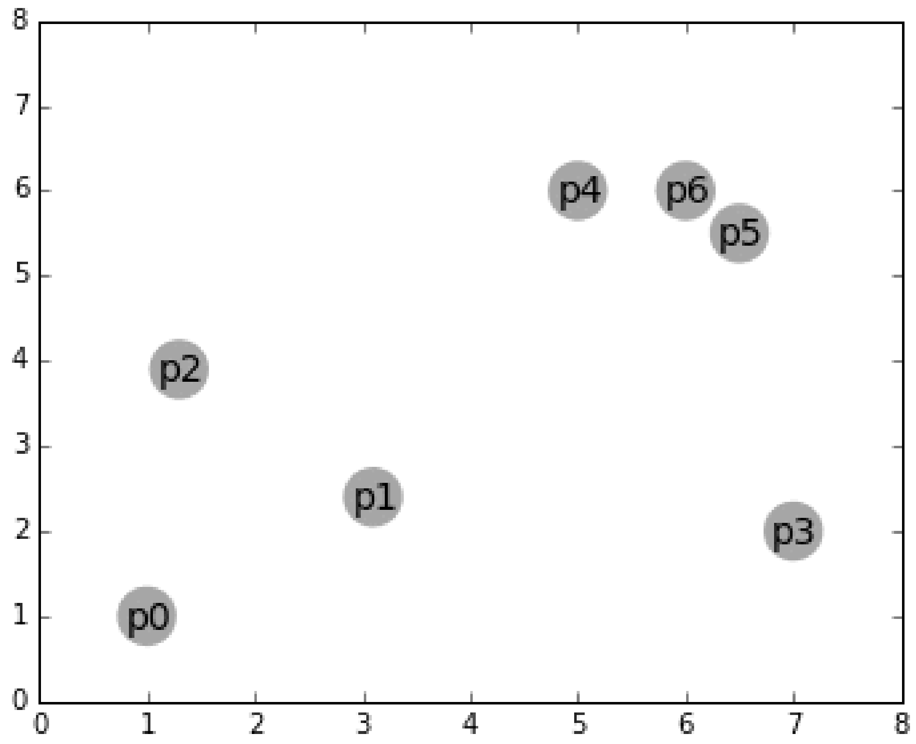
Hierarchical clustering

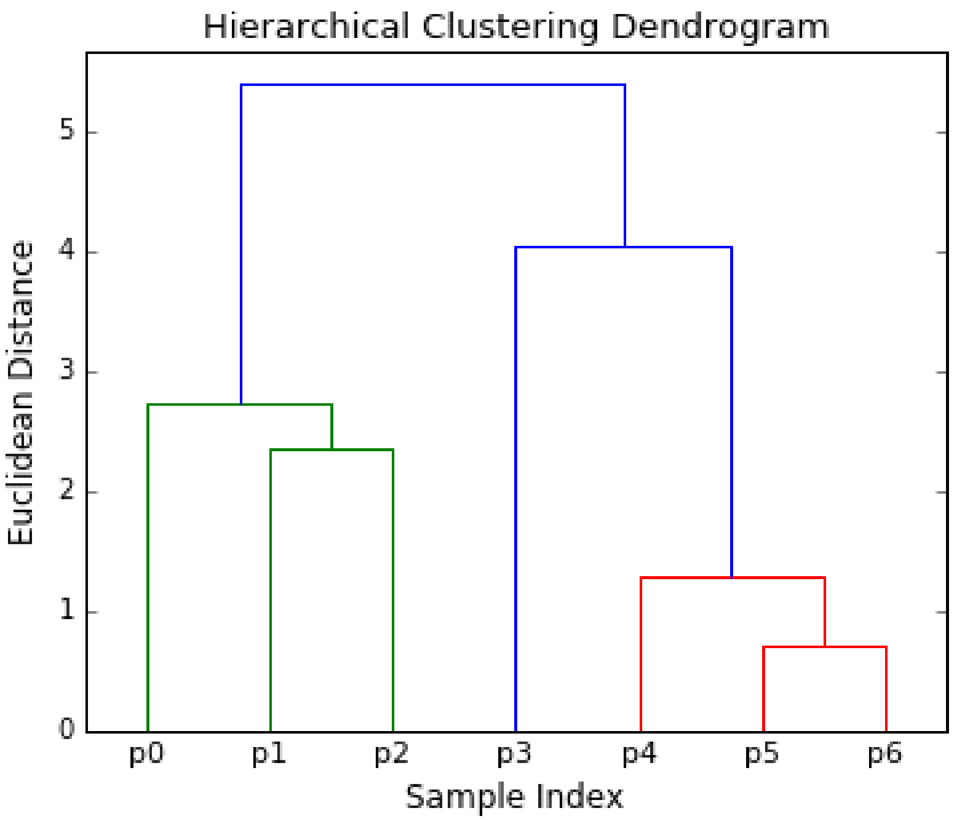
- Dendrogram
  - A tree that shows how clusters are merged/split hierarchically
  - Each node on the tree is a cluster; each leaf node is a singleton cluster
  - A clustering is obtained by cutting the dendrogram at the desired level (then each connected component forms a cluster)



Hierarchical clustering

- Example





Hierarchical clustering

- Three different distance measures
  - Single-nearest distance: single linkage
  - Complete-farthest distance: complete linkage
  - Average distance: average linkage

1) Start by assigning each object to a cluster $c_i = s_i (i = 1, ..., N)$, so that if you have N objects, you have N clusters $\ell = \{c_1, c_2, ..., c_N\}$, each containing just one item.

2) Find the pair of clusters (c_i, c_j) such that $D(c_i, c_j) \leq D(c_{i'}, c_{j'})$, $\forall c_{i'} \neq c_{j'} \in \ell$ and merge them into a single cluster $c_k = c_i \cup c_j$. Delete c_i and c_j from ℓ and insert c_k into ℓ so that now you have one cluster less.

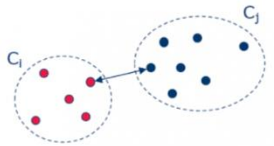
3) Compute distances (similarities) between the new cluster and each of the old clusters.

4) Repeat steps 2) and 3) until all items are clustered into a single cluster of size N.

Hierarchical clustering

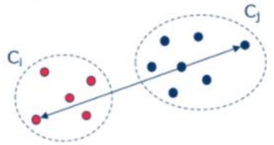
- Three different distance measures
  - Single-nearest distance (single linkage): **shortest** distance between any pair

$$D(c_i, c_j) = \min d(a,b), \forall a \in c_i \text{ and } b \in c_j$$



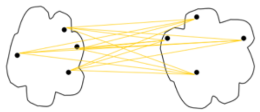
  - Complete-farthest distance (complete linkage): **greatest** distance between any pair

$$D(c_i, c_j) = \max d(a,b), \forall a \in c_i \text{ and } b \in c_j$$



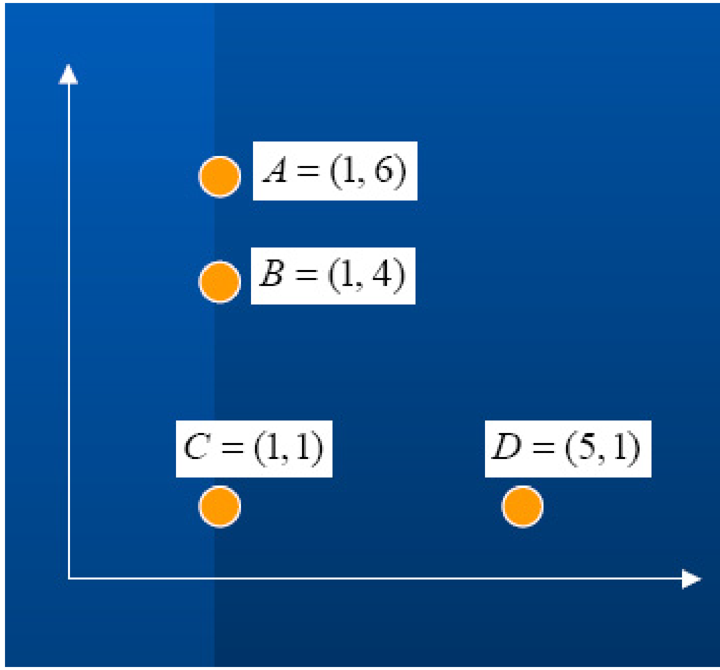
  - Average distance or average linkage: **average** distance between all pairs

$$D(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{a \in c_i, b \in c_j} d(a,b)$$



Hierarchical clustering

- Example: clustering 4 data items in 2D space

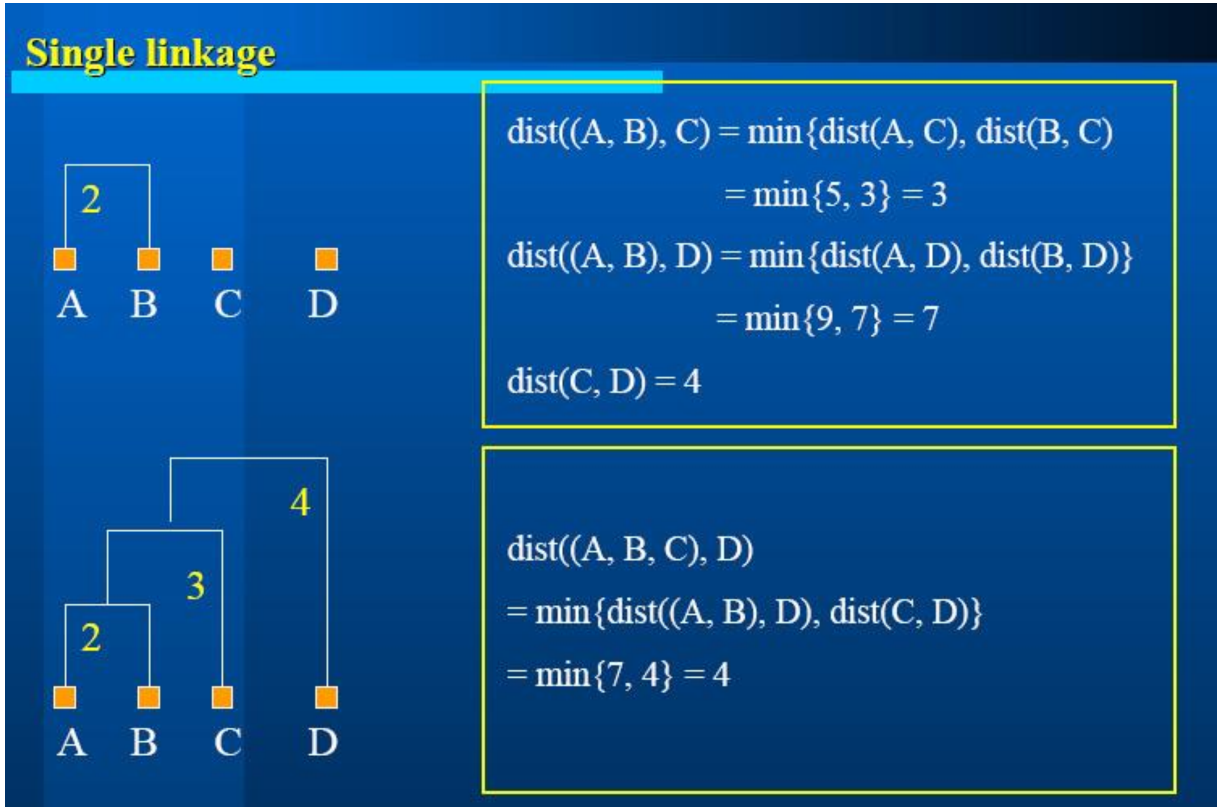


Manhattan distance

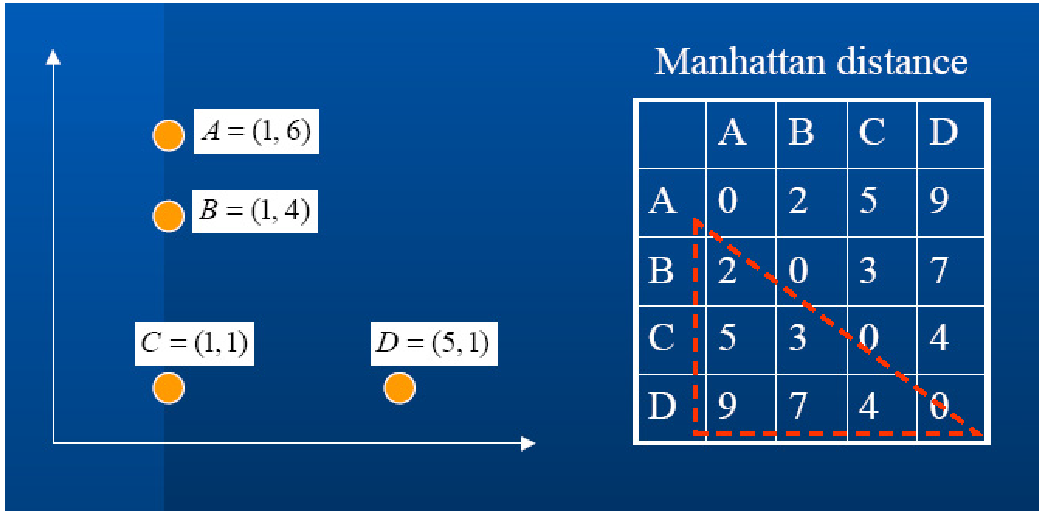
	A	B	C	D
A	0	2	5	9
B	2	0	3	7
C	5	3	0	4
D	9	7	4	0

Hierarchical clustering

- Method: *single-linkage* clustering



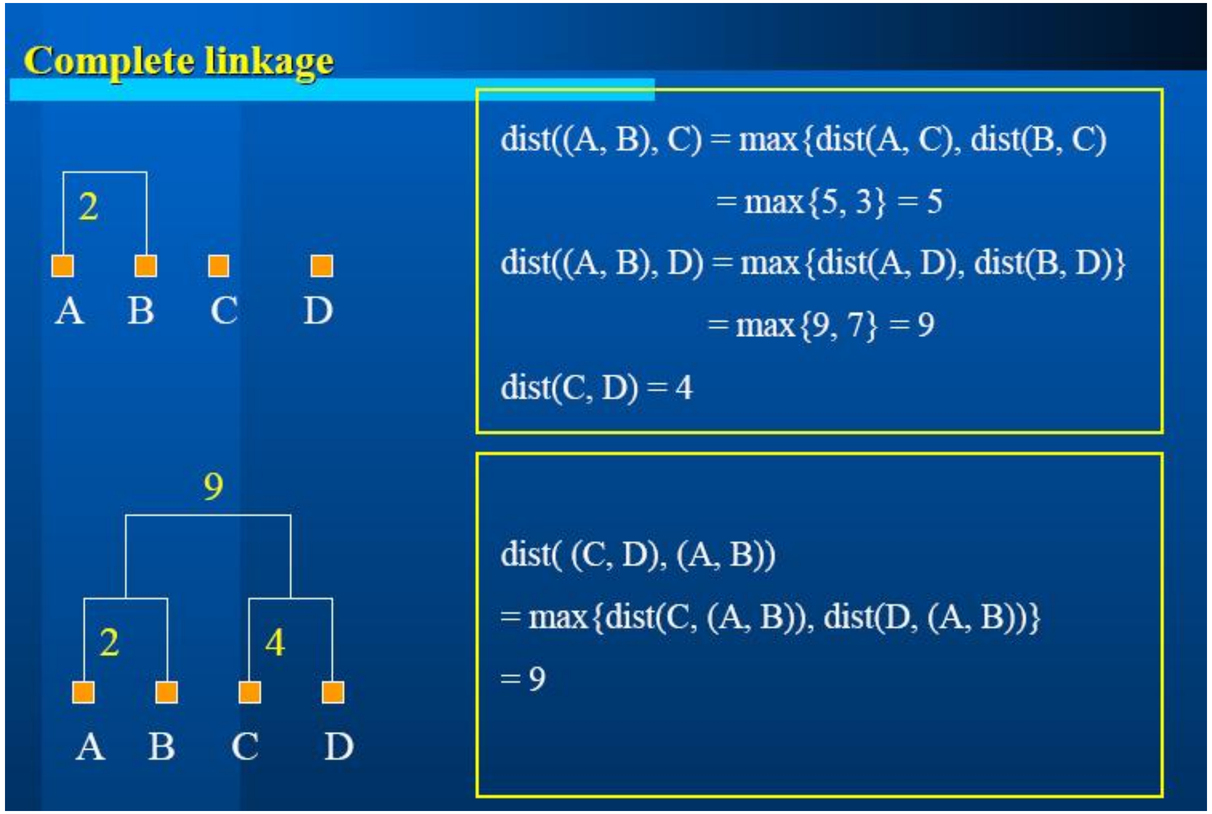
$$D(c_i, c_j) = \min d(a, b), \forall a \in c_i \text{ and } b \in c_j$$



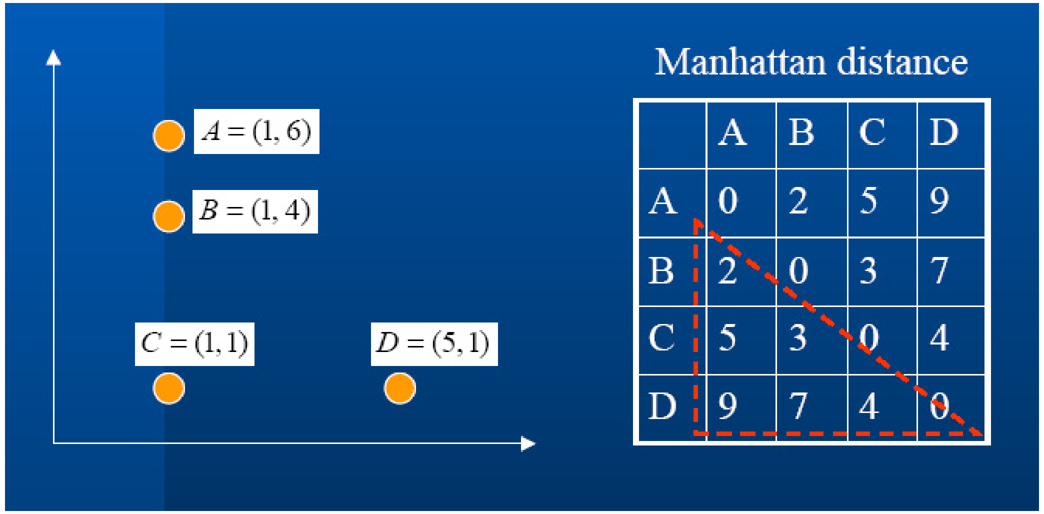
	A	B	C	D
A	0	2	5	9
B	2	0	3	7
C	5	3	0	4
D	9	7	4	0

Hierarchical clustering

- Method: *complete-linkage* clustering



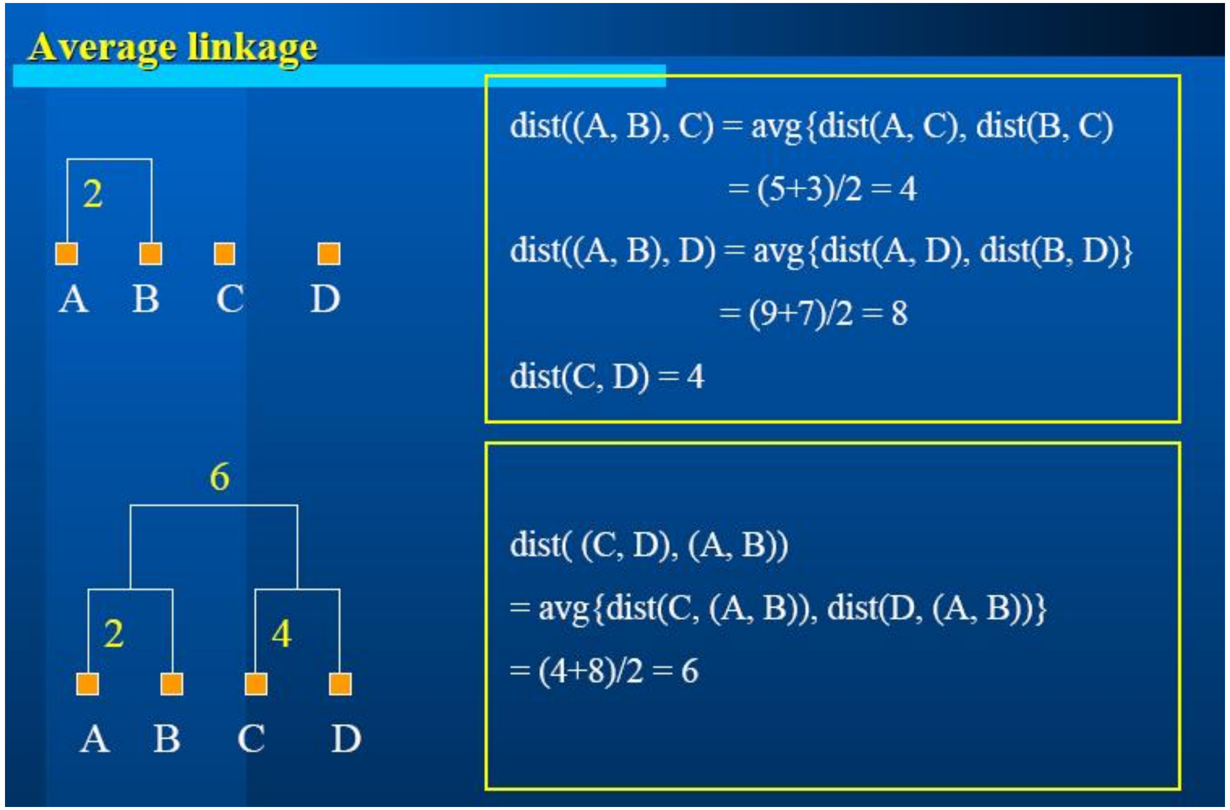
$$D(c_i, c_j) = \max d(a, b), \forall a \in c_i \text{ and } b \in c_j$$



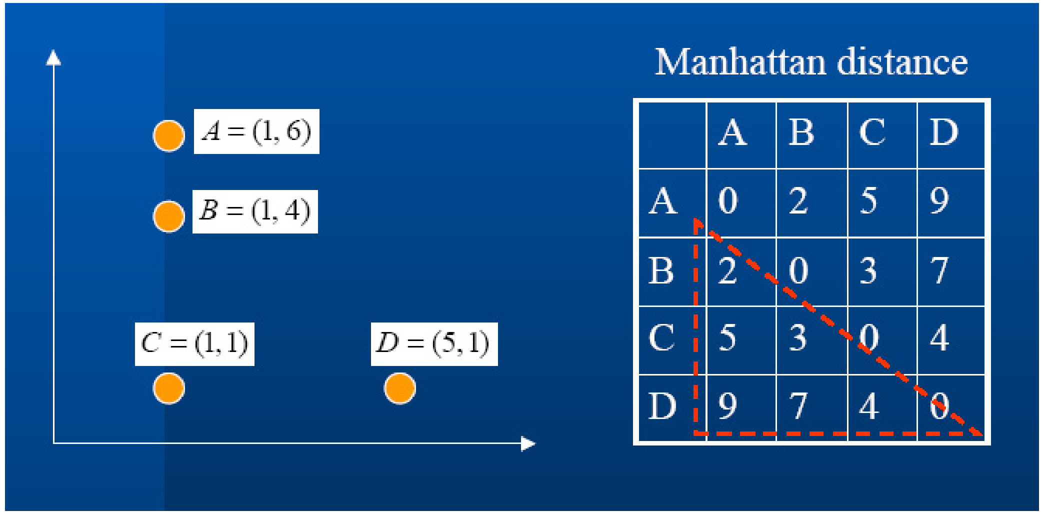
	A	B	C	D
A	0	2	5	9
B	2	0	3	7
C	5	3	0	4
D	9	7	4	0

Hierarchical clustering

- Method: *average-linkage* clustering



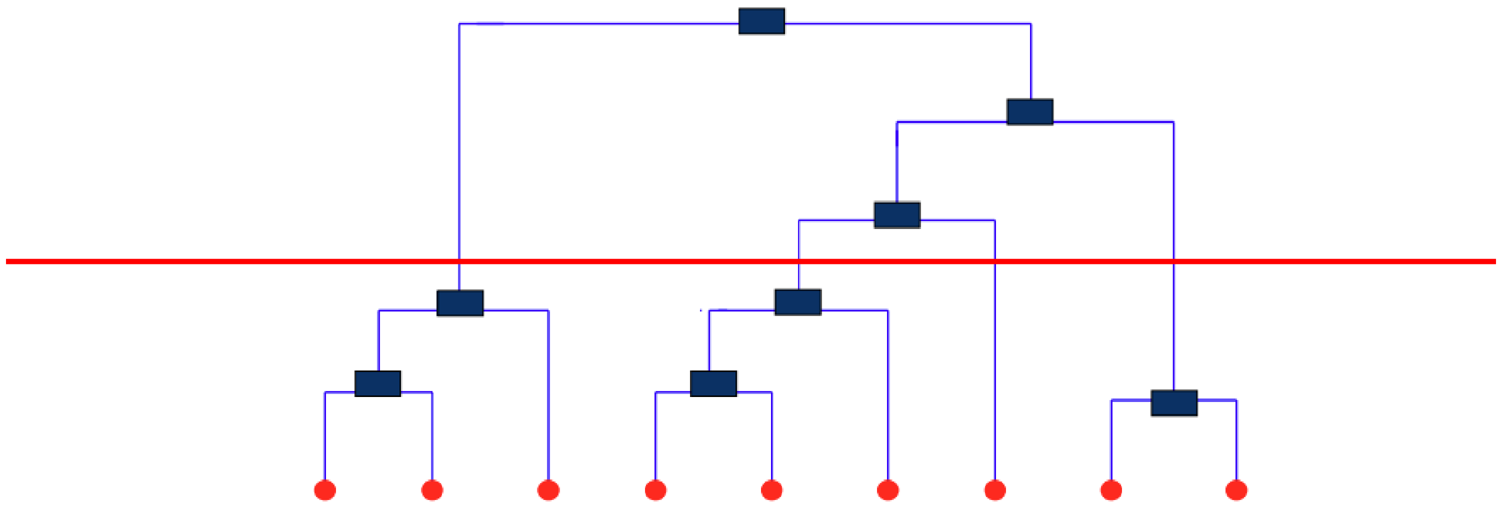
$$D\left(c_i, c_j\right) = \frac{1}{|c_i||c_j|} \sum_{a\in c_i, b\in c_j} d(a,b)$$



	A	B	C	D
A	0	2	5	9
B	2	0	3	7
C	5	3	0	4
D	9	7	4	0

Hierarchical clustering

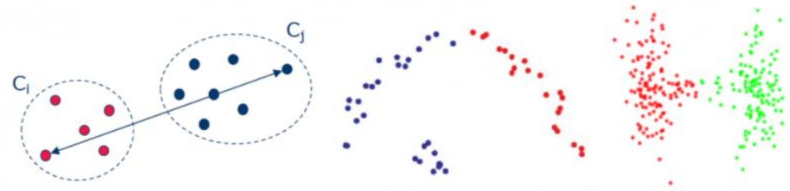
- Advantages
  - No a priori information about the number of clusters required
    - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
  - Easy to implement and gives best result in some cases



Hierarchical clustering

- Advantages
  - No a priori information about the number of clusters required
  - Easy to implement and gives best result in some cases
- Limitations
  - Can never undo what (i.e., merging two clusters) was done previously
  - Can be slow if a large number data points (due to pairwise distance computation)
  - It may not be easy to choose a proper distance measure

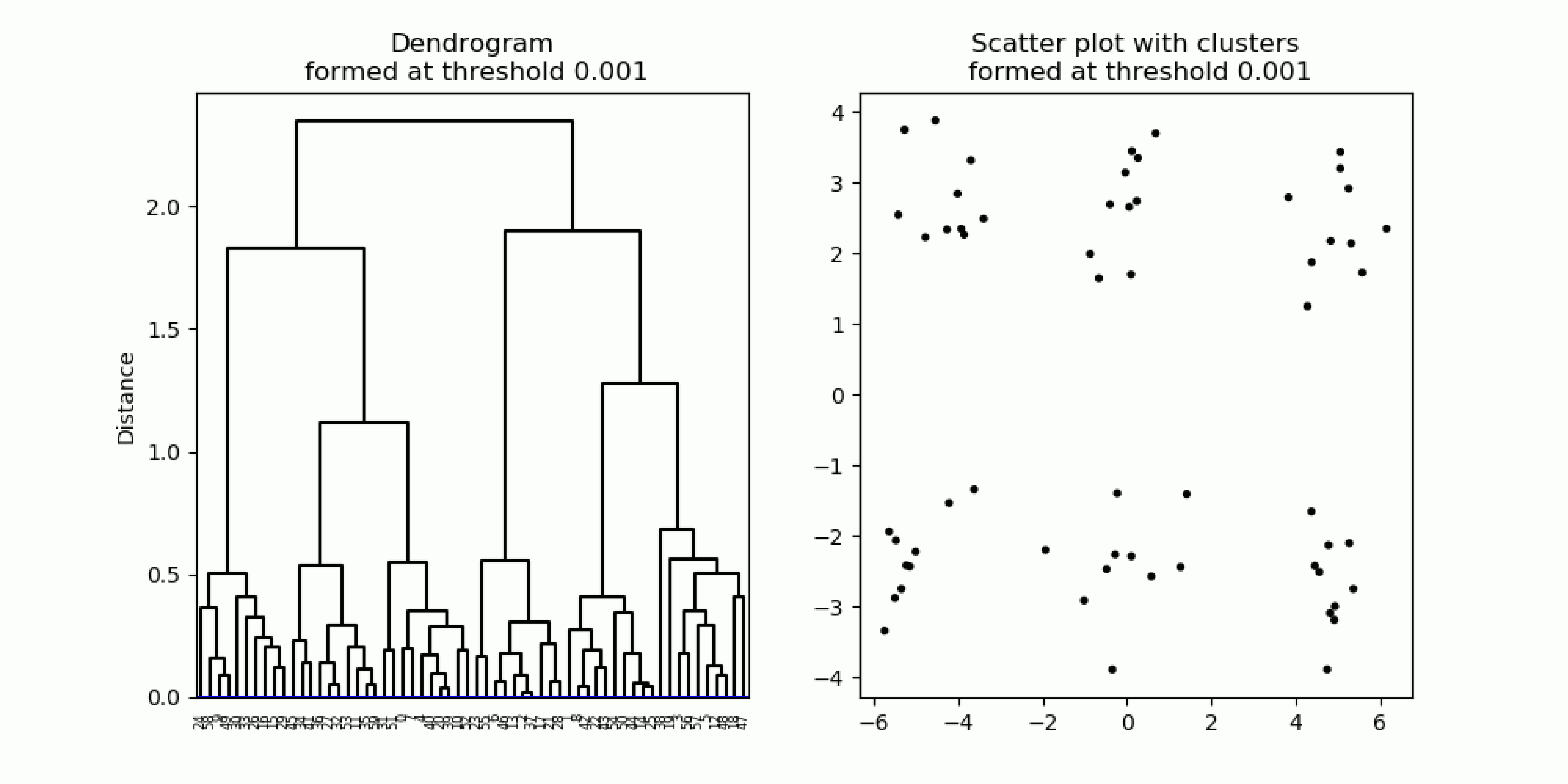
Single linkage: well-separated clusters



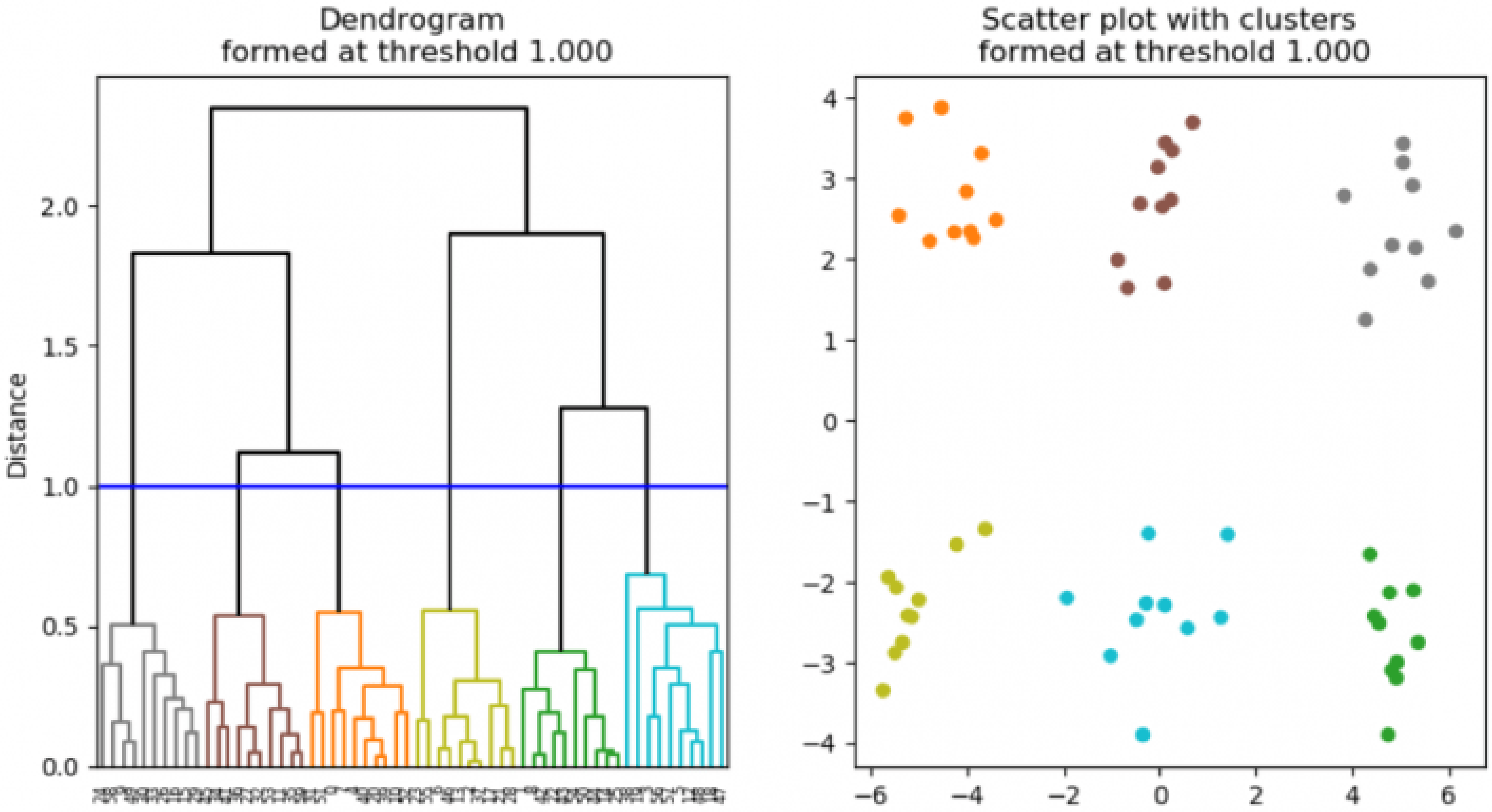
Complete linkage: compact clusters

  - It may not be easy to identify the correct number of clusters by the dendrogram

Hierarchical clustering

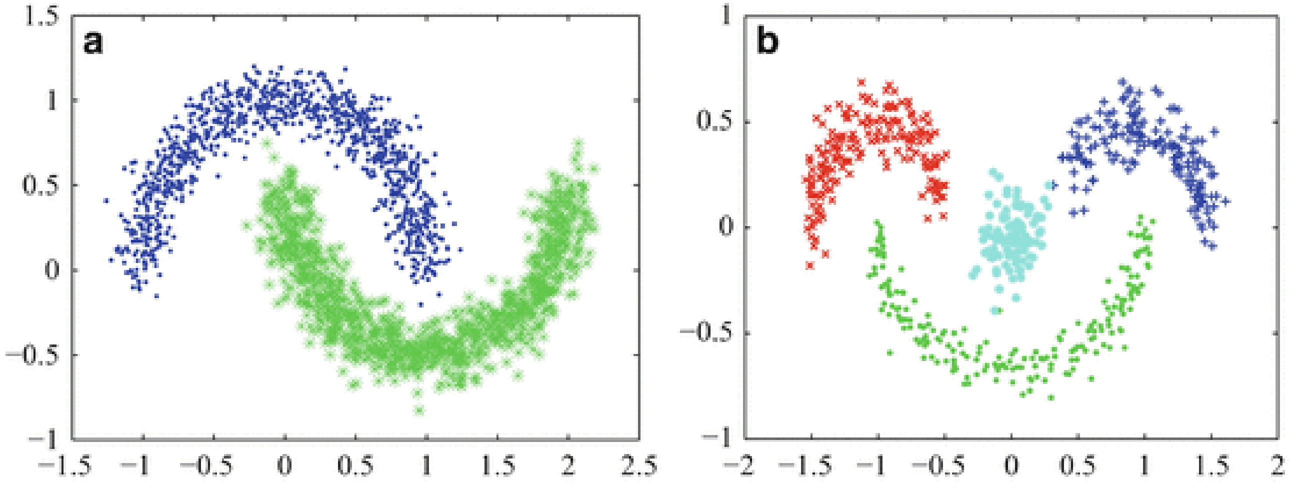


Hierarchical clustering



Hierarchical clustering

- Can hierarchical clustering method handle?

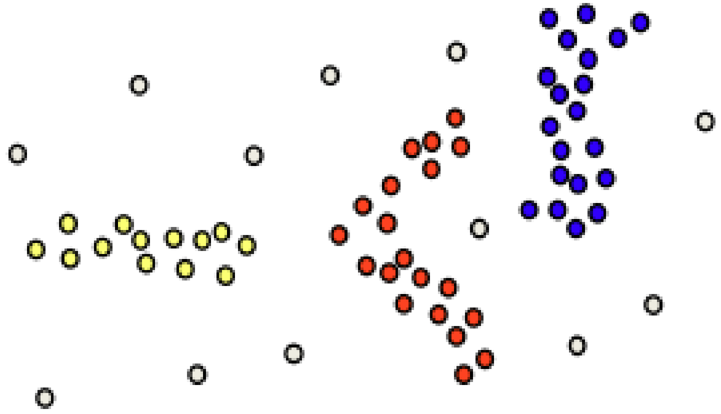


Agenda

- Overview
  - What is clustering?
  - Distance measure (similarity measure)
  - Types of clustering algorithms
- Clustering algorithms
  - K-means clustering
  - Hierarchical clustering
  - Density-based clustering
- Nearest neighbor classification
- Features

Density-based clustering

- Basic ideas
  - Clusters are contiguous regions of high density in the data space, separated by regions of lower data density
  - A cluster is defined as a maximal set of density connected points
- DBSCAN
  - Density-Based Spatial Clustering of Applications with Noise

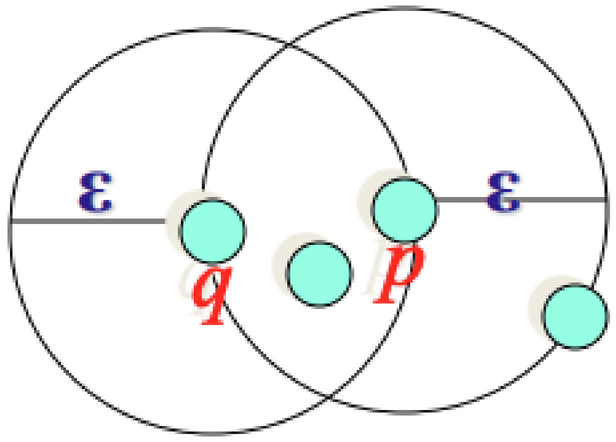


Density definition: two parameters

- ϵ-neighborhood: objects within a radius of ϵ from a cluster

$$N_{\varepsilon}(p): \{q \mid d(p,q) \le \varepsilon\}$$

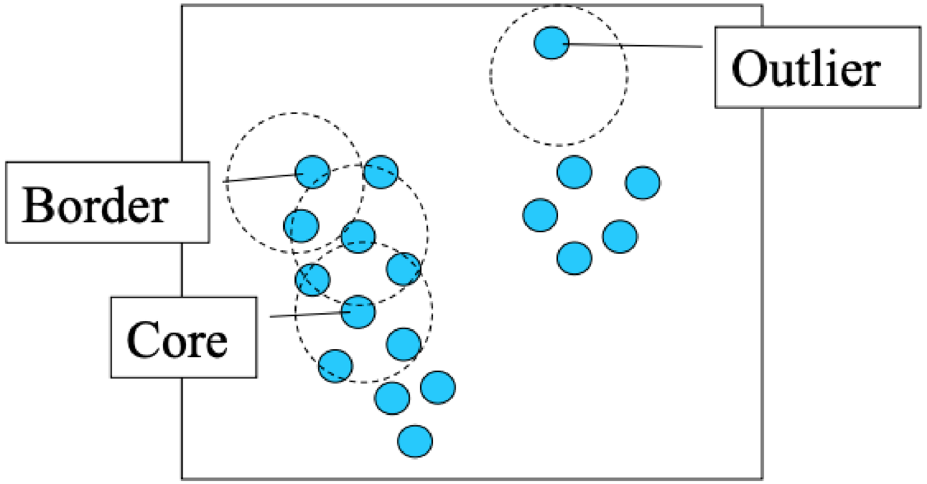
- The minimum number of points required to form a cluster
  - High density: ϵ-neighborhood of an object contains at least *minPts* of objects.



ϵ-neighborhood of *p* and *q*

Three types of data points

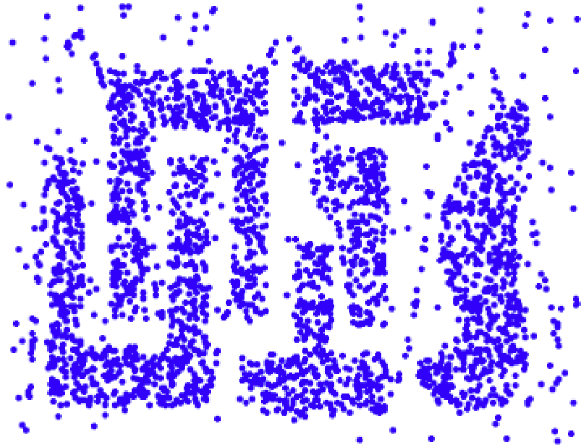
- Given ϵ and *minPts*
    - Core point: has at least *minPts* neighbors within its ϵ-neighborhood
        - At the interior of a cluster
    - Border point
        - has fewer than *minPts* neighbors within its ϵ-neighborhood
        - is within the ϵ-neighborhood of a core point
    - Outlier/Noise
        - Any point that is neither core nor border



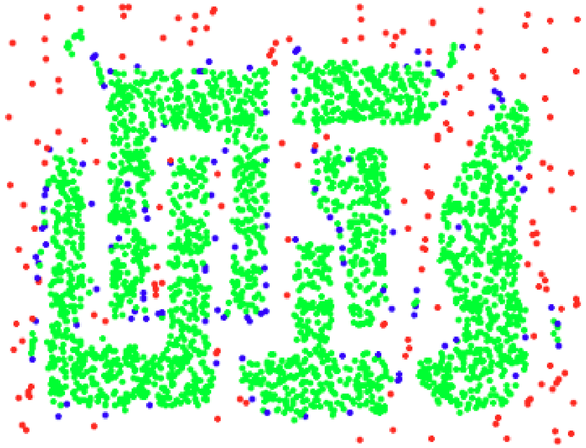
ϵ = circle radius, *minPts* = 5

Three types of data points

- Example



Original Points

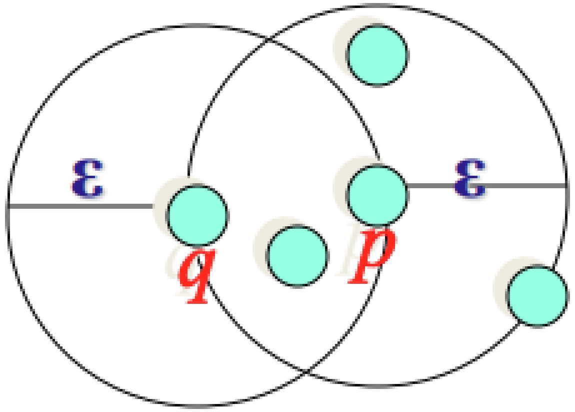


ϵ = 10, *minPts* = 4

Point types: core, border and outliers

Density definition: two concepts

- Density reachability
  - A point *q* is said to be directly reachable from a point *p* if
    - ***p* is a core point** (i.e., has at least *minPts* points within ϵ-neighborhood)
    - *q* is within distance ϵ from *p*

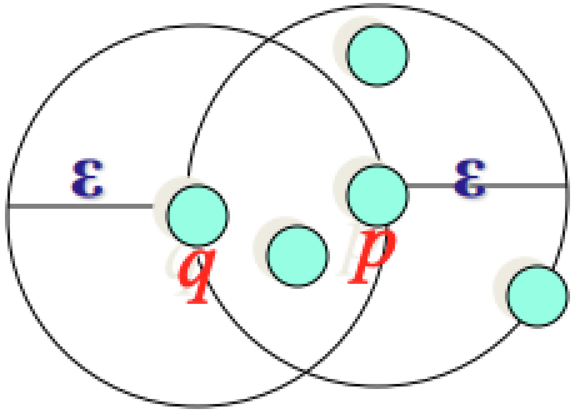


minPts = 4

In this example, *q* is density reachable from *p*.
Is *p* also density reachable from *q*?

Density definition: two concepts

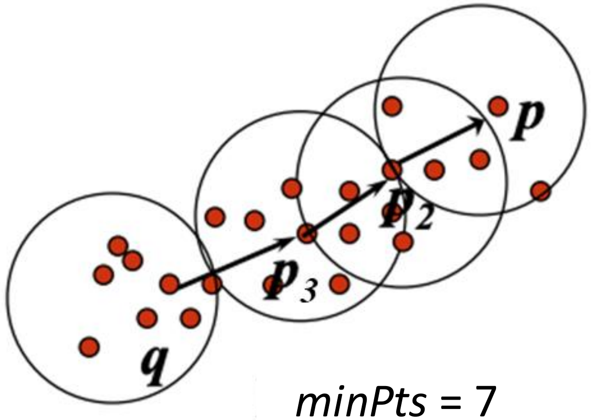
- Density reachability
  - A point *q* is said to be directly reachable from a point *p* if
    - ***p* is a core point** (i.e., has at least *minPts* points within ϵ-neighborhood)
    - *q* is within distance ϵ from *p*
  - Density reachability is asymmetric
    - Only core points can reach other points



minPts = 4

Density definition: two concepts

- Density reachability
- Density connectivity
  - A point p is said to be reachable from q if
    - There is a path $p_1, \ldots, p_n$ with $p_1 = p$ and $p_n = q$, where each p_i is directly reachable from p_{i+1}
  - Density connectivity is transitive (i.e., it forms a chain)



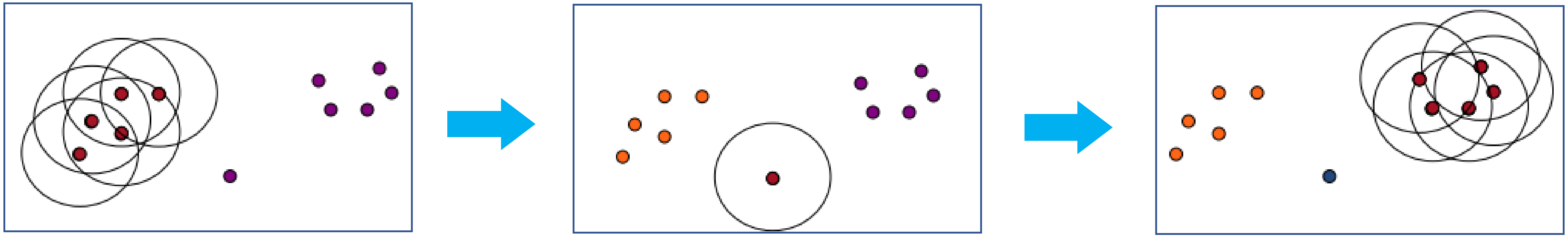
minPts = 7

Example:
- p is directly reachable from p_2
- p_2 is directly reachable from p_3
- p_3 is directly reachable from q

So we say: p is reachable from q (p *and* q density connected)

DBSCAN algorithm

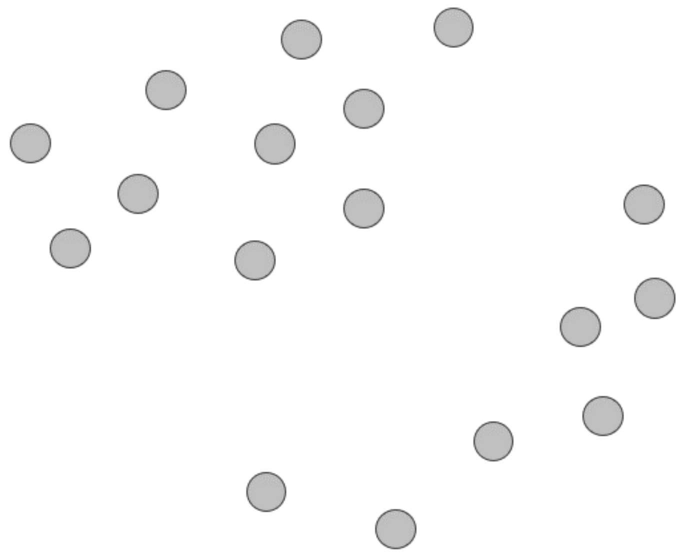
for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-connected by *o*,
      and assign them to a new cluster.
    else
      assign o to NOISE



An example of DBSCAN clustering: *minPts* = 3

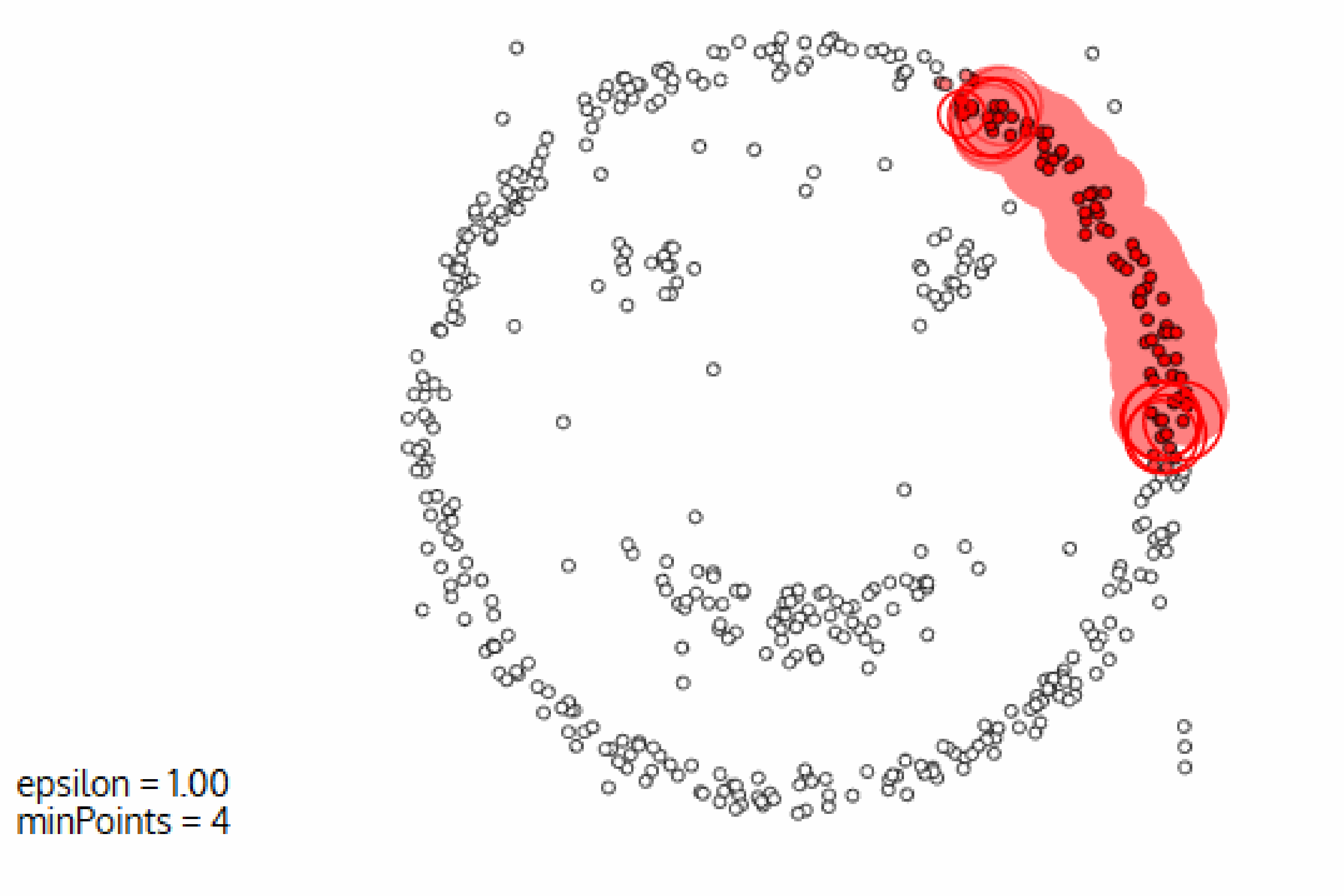
DBSCAN algorithm

- Illustration



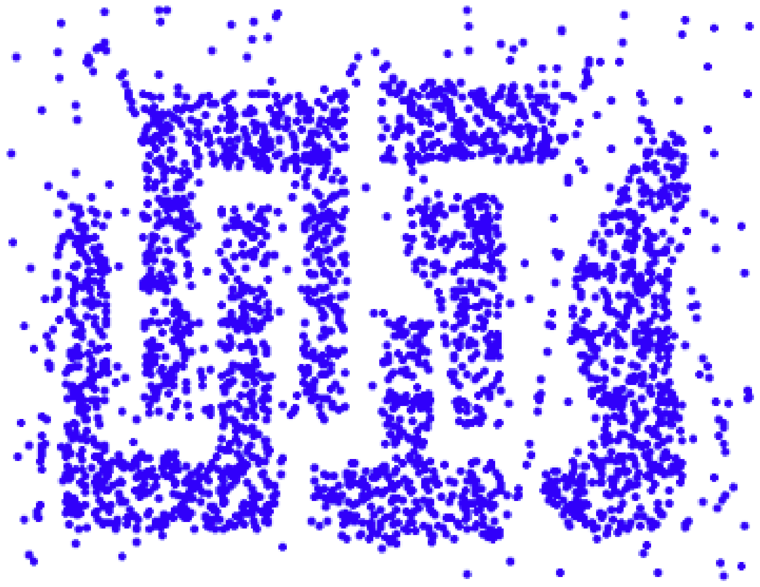
DBSCAN algorithm

- Illustration

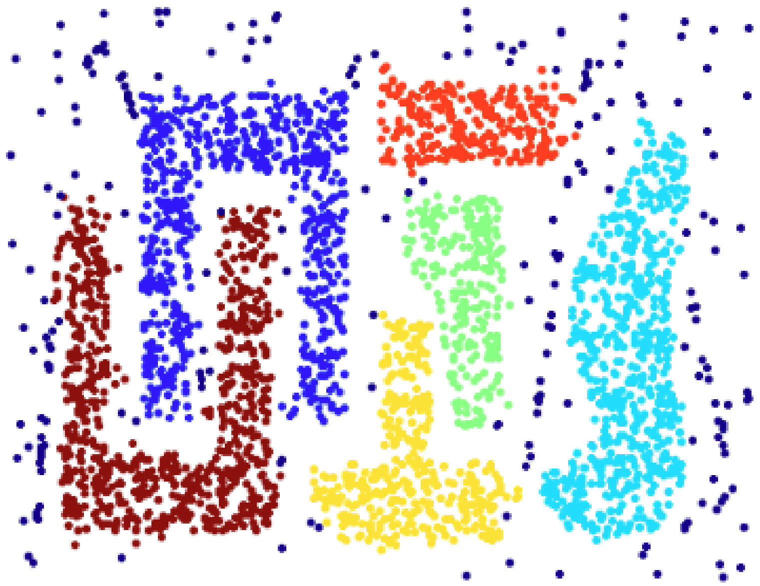


DBSCAN algorithm

- Example



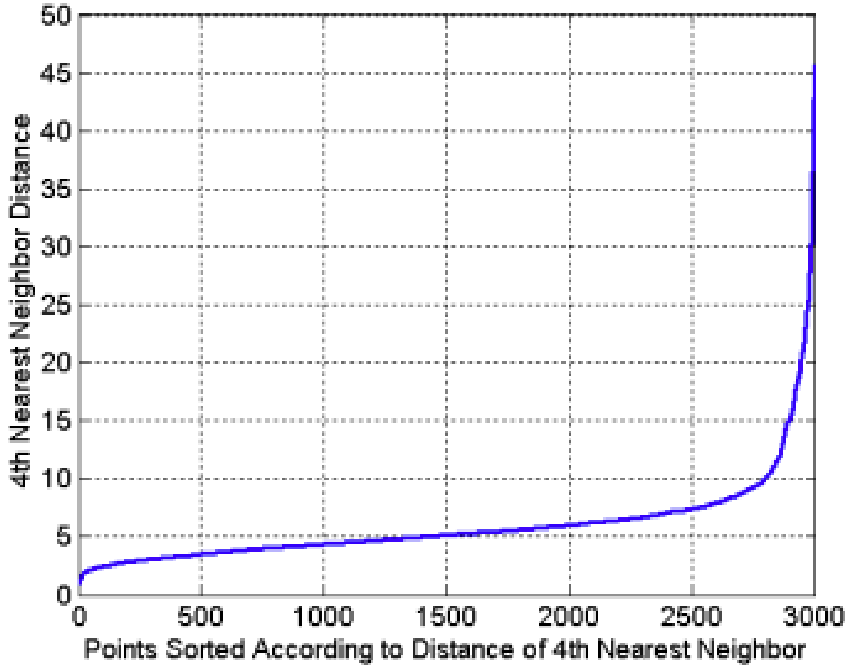
Original Points



Clusters

DBSCAN algorithm

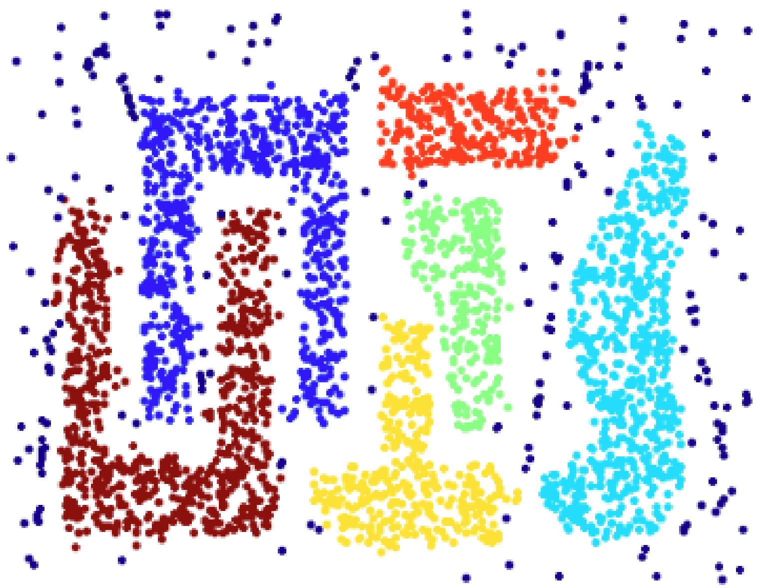
- Determining the two parameters
  - *minPts*
    - *minPts* = 1?
    - *minPts* = 2 (same as single linkage hierarchical method, with dendrogram cut at height ε)
    - *minPts* = 2 * dimension
  - ϵ (distance threshold)
    - k-distance graph ($k = minPts - 1$)
    - Look for the “elbow”
      - ➢ The point with maximum curvature



K-distance graph plots the k-th nearest neighbor distance of all points sorted from smallest to largest.

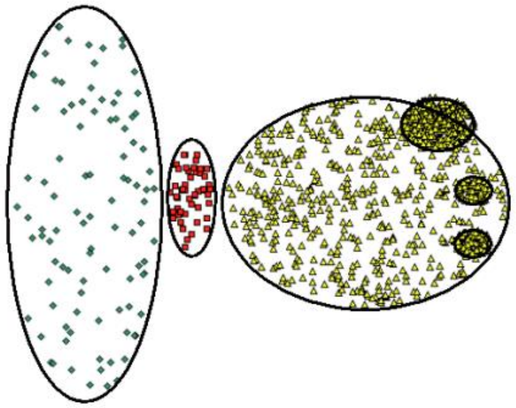
DBSCAN algorithm

- Advantages
  - Resistant to Noise
  - Robust to clusters of different shapes and sizes

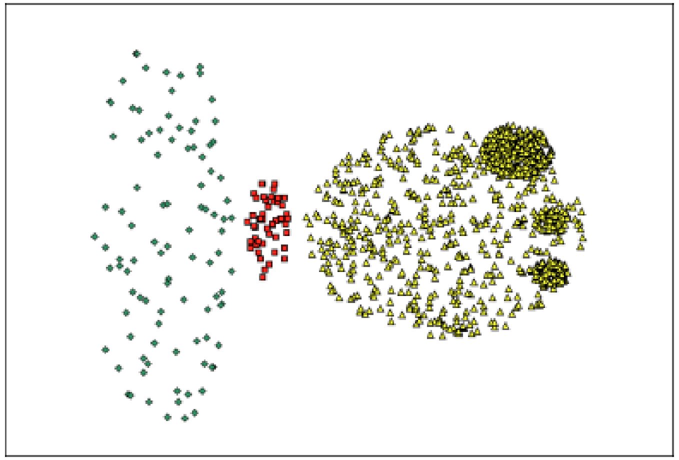


DBSCAN algorithm

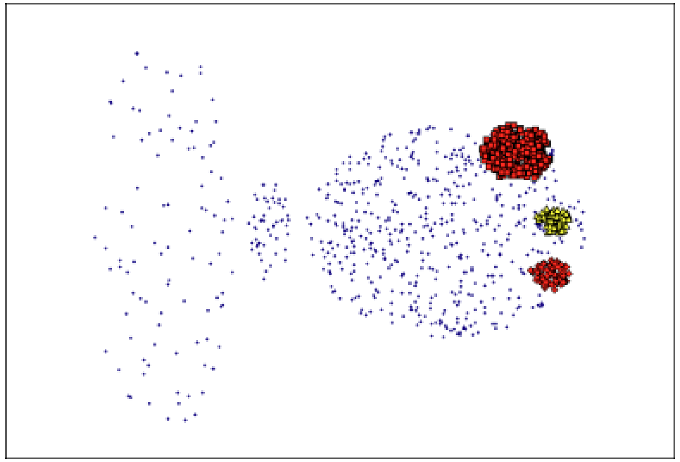
- Limitations
  - Cannot handle varying densities
  - Hard to determine a good set of parameters



Original points



minPts = 4, ϵ = 9.92



minPts = 4, ϵ = 75

Question

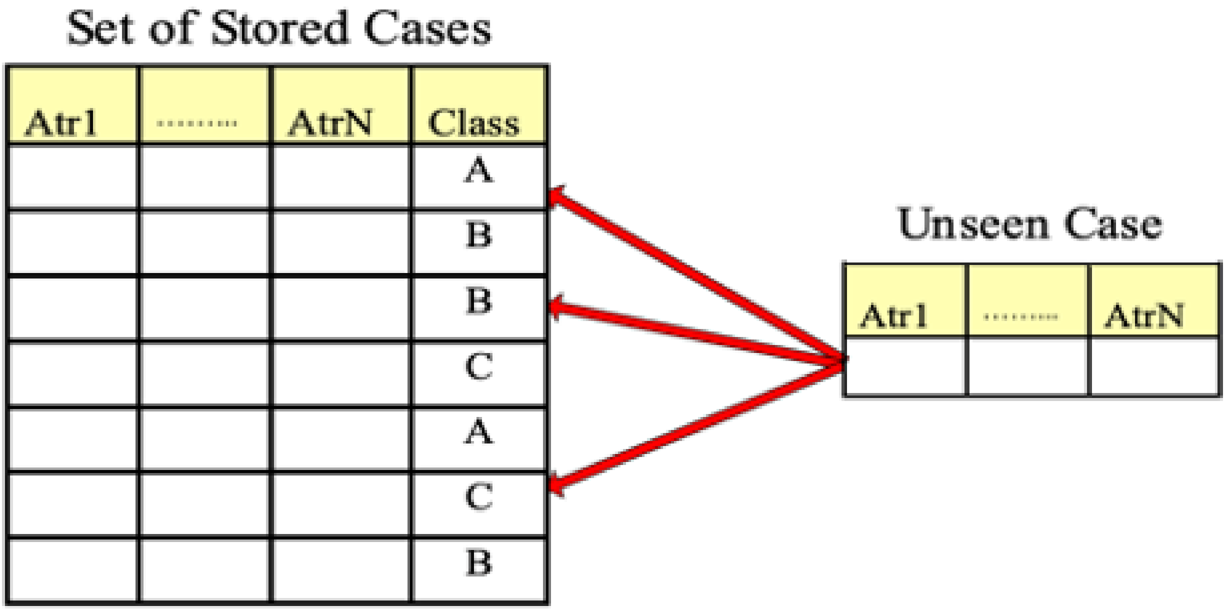
- Which method (DBSCAN or k-means) was used to produce each result?

Agenda

- Overview
    - What is clustering?
    - Distance measure (similarity measure)
    - Types of clustering algorithms
- Clustering algorithms
    - K-means clustering
    - Hierarchical clustering
    - Density-based clustering
- Nearest neighbor classification
- Features

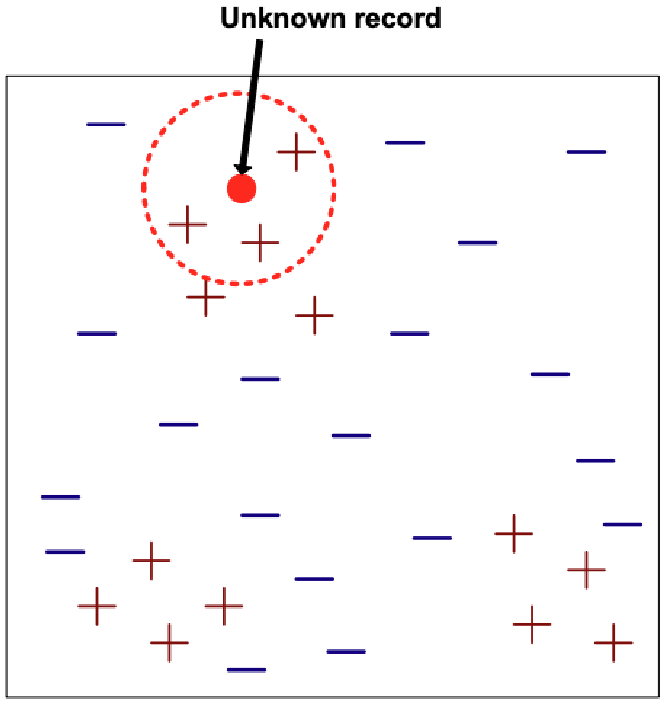
Nearest neighbor classification

- Basic ideas
  - Store the training records
  - Use training records to predict the class label of unseen cases



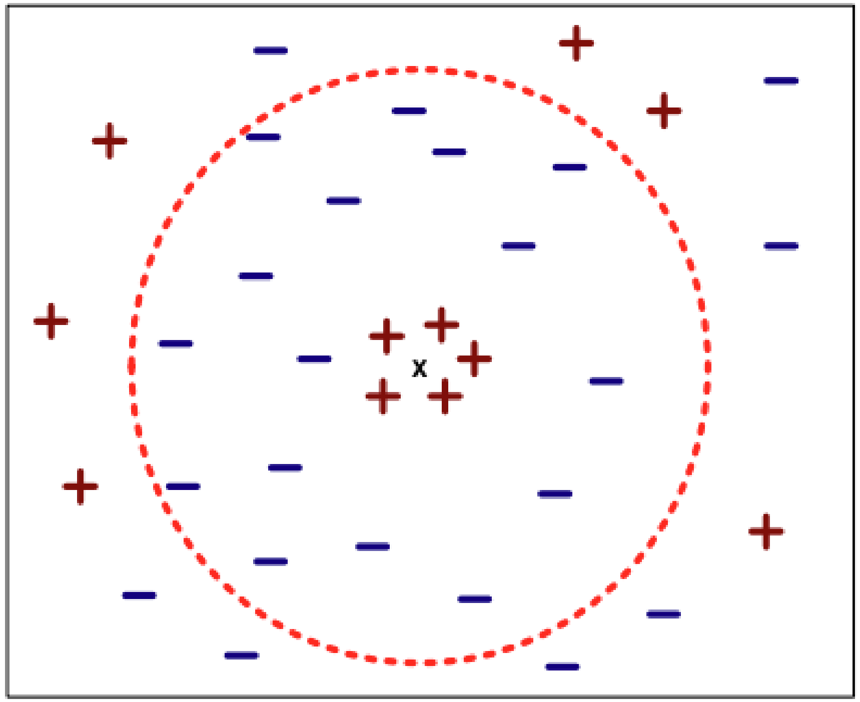
Nearest neighbor classification

- Requires three things
  - The set of stored records
  - Distance metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve
- To classify an unknown record
  - Compute distance to other training records
  - Identify k nearest neighbors
  - Use class labels of the k nearest neighbors to determine the class label of the unknown record (e.g., by taking majority vote)



Nearest neighbor classification

- Choosing the value of k
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



Agenda

- Overview
  - What is clustering?
  - Distance measure (similarity measure)
  - Types of clustering algorithms
- Clustering algorithms
  - K-means clustering
  - Hierarchical clustering
  - Density-based clustering
- Nearest neighbor classification
- Features

Features

- A set of attributes of an object
- Typically stored as a vector – feature vector
- Scaling issue: distance measure dominated by one of the attributes
  - Example
    - height of a person [1.5m, 1.8m]
    - weight of a person [40kg, 100kg]
    - income of a person [€10K, €1M]
  - Solution
    - Normalization, i.e., $\frac{\text{each attribute value}}{\text{max possible value of this attribute}}$

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

You should have learned

- Clustering
  - The basic ideas, strengths, and weaknesses of the 3 clustering methods
  - K-means
    - How is K-means interpreted as an optimization problem?
  - Hierarchical clustering
    - Several ways of defining inter-cluster distance
  - Density-based clustering
    - The parameters and the definitions of neighborhood and density in DBSCAN
- Classification
  - The basic idea of k-nearest neighbor classifier

Next Lecture

- Bayesian classification & logistic regression