

# Data Mining: Concepts and Techniques

(3<sup>rd</sup> ed.)

— Chapter 10 —

Slides Courtesy of Textbook

# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- ~~Grid-Based Methods~~
- Evaluation of Clustering
- Summary

# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary



A CEO wants to re-structure his cooperation

- Want to maximize collaboration and minimize effort
  - Similar employees are in the same department
  - Dissimilar employees are in different departments
- Can the CEO use classification to group the customers?
  - Labeling is expensive: requires extensive labors
  - Labeling is hard: requires extensive analysis to label
- Can we group customers by automatically analyzing the patterns in the customer data?
  - Cluster Analysis



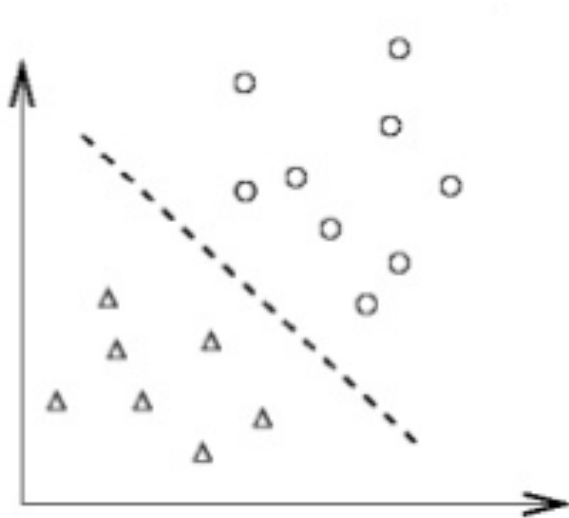
# What is Cluster Analysis?

- Cluster: A collection of data objects:
  - similar (or related) to one another within the same cluster
  - dissimilar (or unrelated) to the objects in other clusters
- Cluster analysis (or clustering, data segmentation...):
  - Given a set of data points, partition them into a set of groups (i.e., clusters) which are as similar as possible
- Cluster analysis is unsupervised learning (i.e., no predefined classes)
  - This contrasts with classification (i.e., supervised learning)
- Typical ways to use cluster analysis:
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing/intermediate step for other algorithms

# Classification vs. Clustering

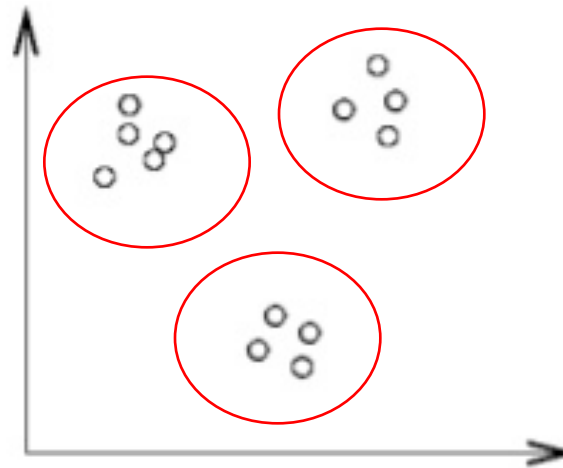
## Classification

- Labeled data points
- Learn rules to assign labels to data points
- Supervised learning



## Clustering

- Unlabeled data points
- Group data points based on data patterns/structures
- Unsupervised learning



# Applications of Cluster Analysis

- A key intermediate step for other data mining tasks
  - Generating a compact summary of data for classification, pattern discovery, hypothesis generation and testing, etc.
  - Outlier detection: Outliers—those “far away” from any cluster
- Data summarization, compression, and reduction
  - Ex. Image processing: Vector quantization
- Collaborative filtering, recommendation systems, or customer segmentation
  - Find like-minded users or similar products
- Web Search
  - Clustering search results
- Dynamic trend detection
  - Clustering stream data and detecting trends and patterns
- Multimedia data analysis, biological data analysis and social network analysis
  - Ex. Clustering images or video/audio clips, gene/protein sequences, etc.

# Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)



# Requirements and Challenges

- Quality
  - High **intra-class** similarity: **cohesive** within clusters
  - Low **inter-class** similarity: **distinctive** between clusters
  - Subjective quality measures
- Versatility
  - Ability to deal with different types of attributes: Numerical, categorical, text, multimedia, networks, and mixture of multiple types
  - Ability to discover clusters with arbitrary shape
  - Ability to deal with noisy data
  - Ability to deal with streaming data: insensitivity to input order
- Scalability
  - Large scale
  - High dimensionality
- Consistency with user-given constraints
  - Domain knowledge
  - User queries
- Interpretability and usability

# Basic Steps to Develop a Clustering Task

- Select features
  - Select info with regard to the task of interest
  - Minimize redundancy
- Design proximity measure
  - Measure distance/similarity between two instances, i.e., two feature vectors
- Select clustering criteria
  - Design criteria to cluster instances via a cost function or some rules
- Design clustering algorithms
  - Design road map to cluster all instances
- Validate the results
  - Carry out validation test (also, *clustering tendency* test)
- Interpret the results


# Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

# Major Clustering Approaches (II)

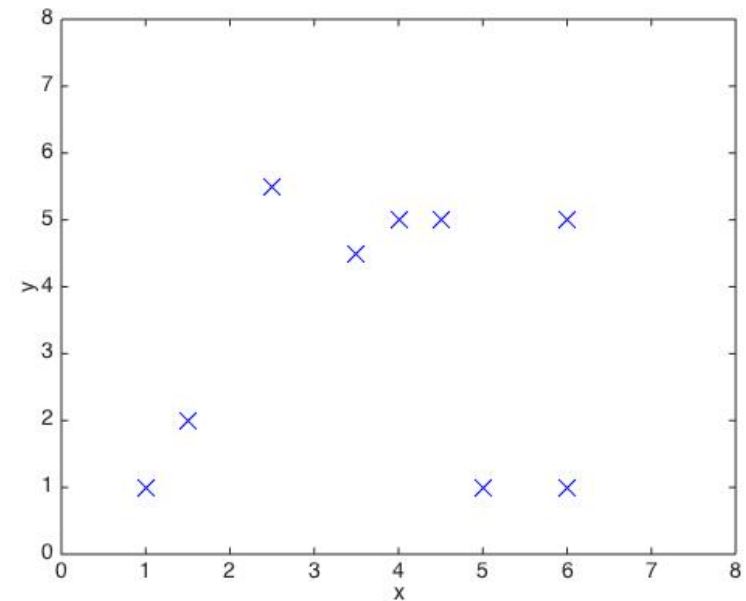
- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

# Chapter 10. Cluster Analysis: Basic Concepts and Methods

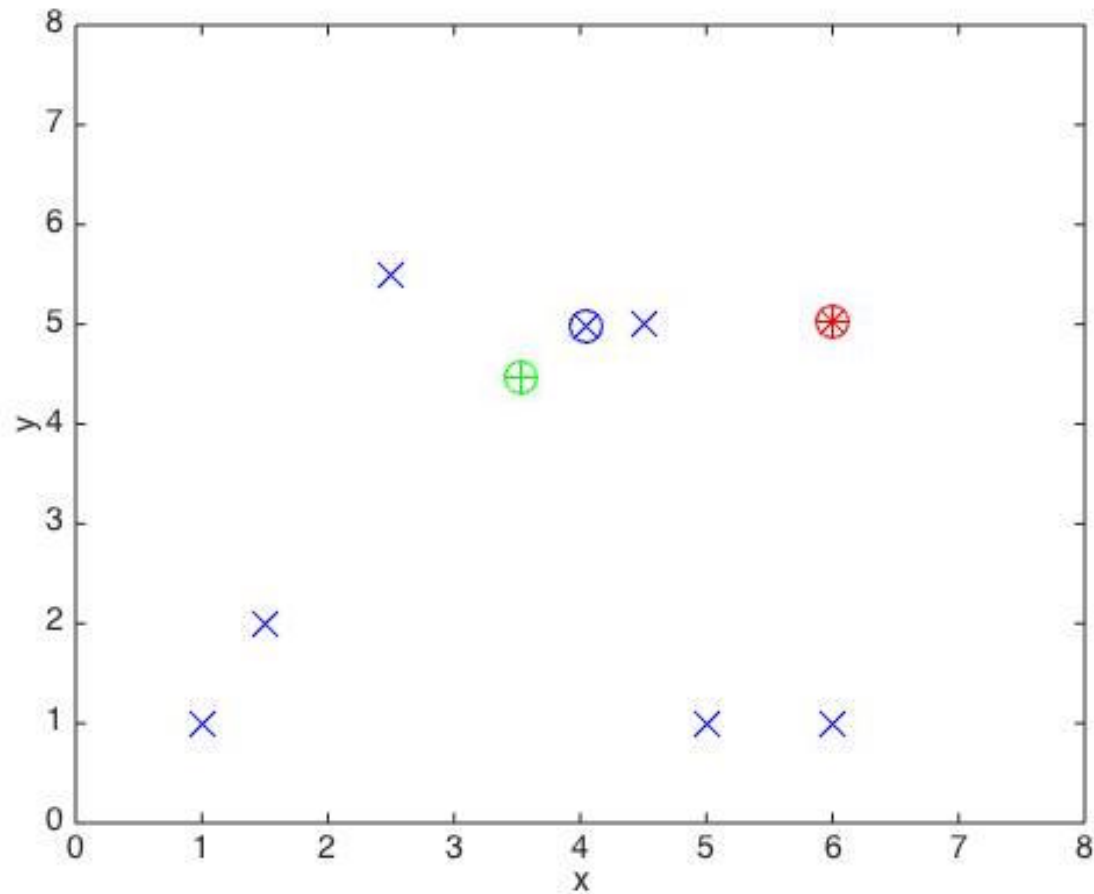
- Cluster Analysis: Basic Concepts
- Partitioning Methods 
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

# Motivating example: 2-D data points

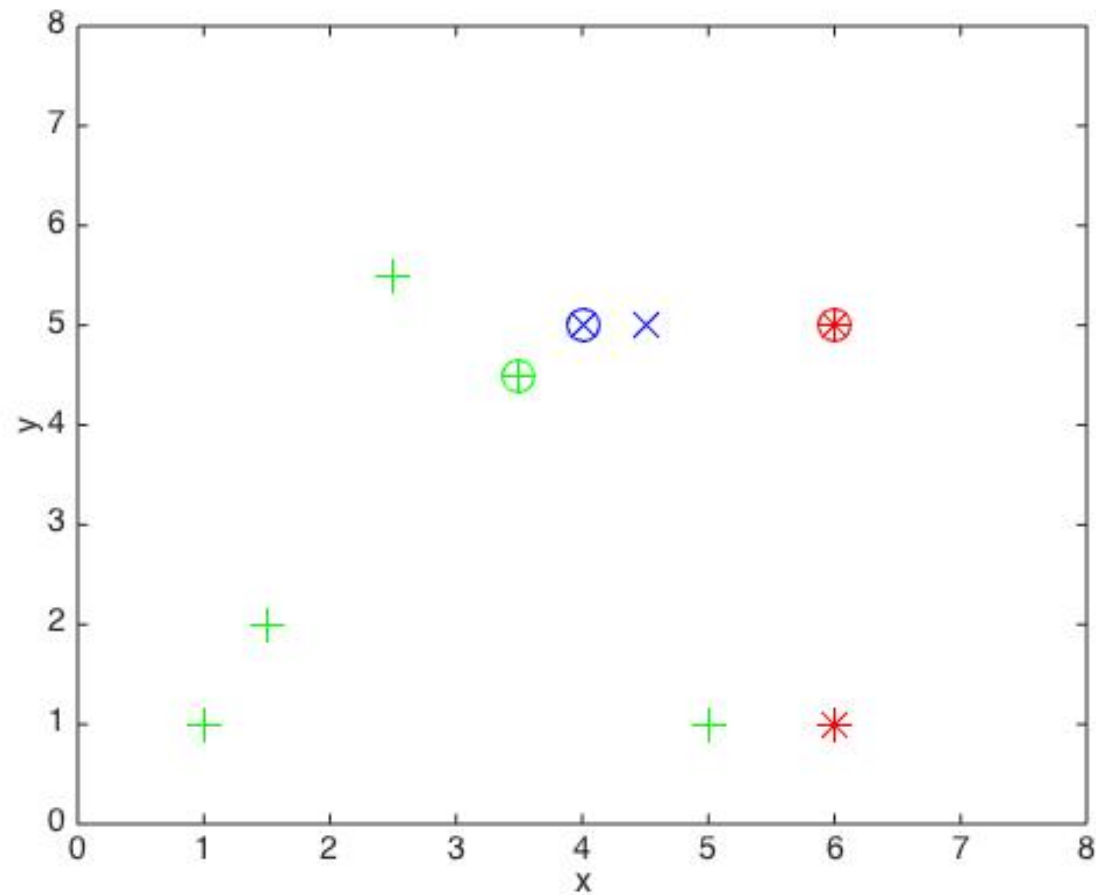
x	y
1	1
1.5	2
2.5	5
6	5
4	5
4.5	5
3.5	4.5
5	1
6	1



How to partition data points into 3 clusters if knowing the centers of those clusters?

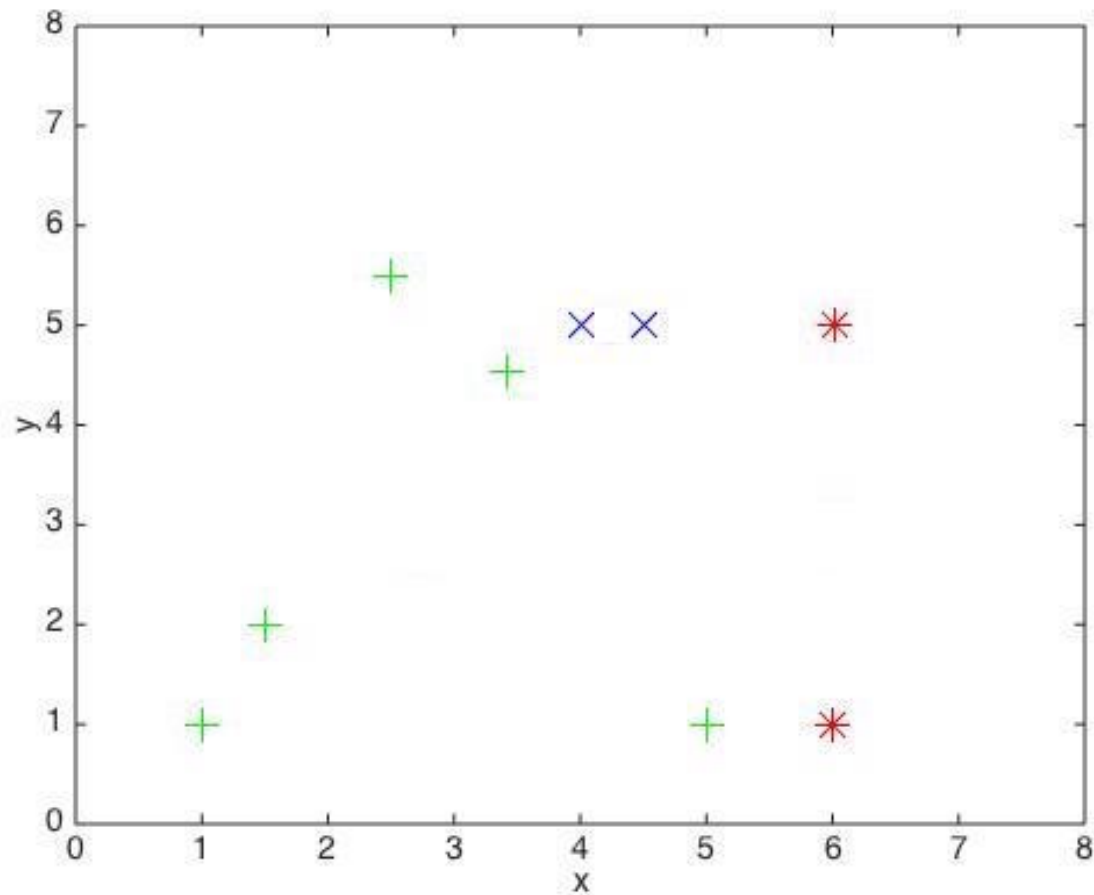


Partitioning by assigning points to closer cluster centers

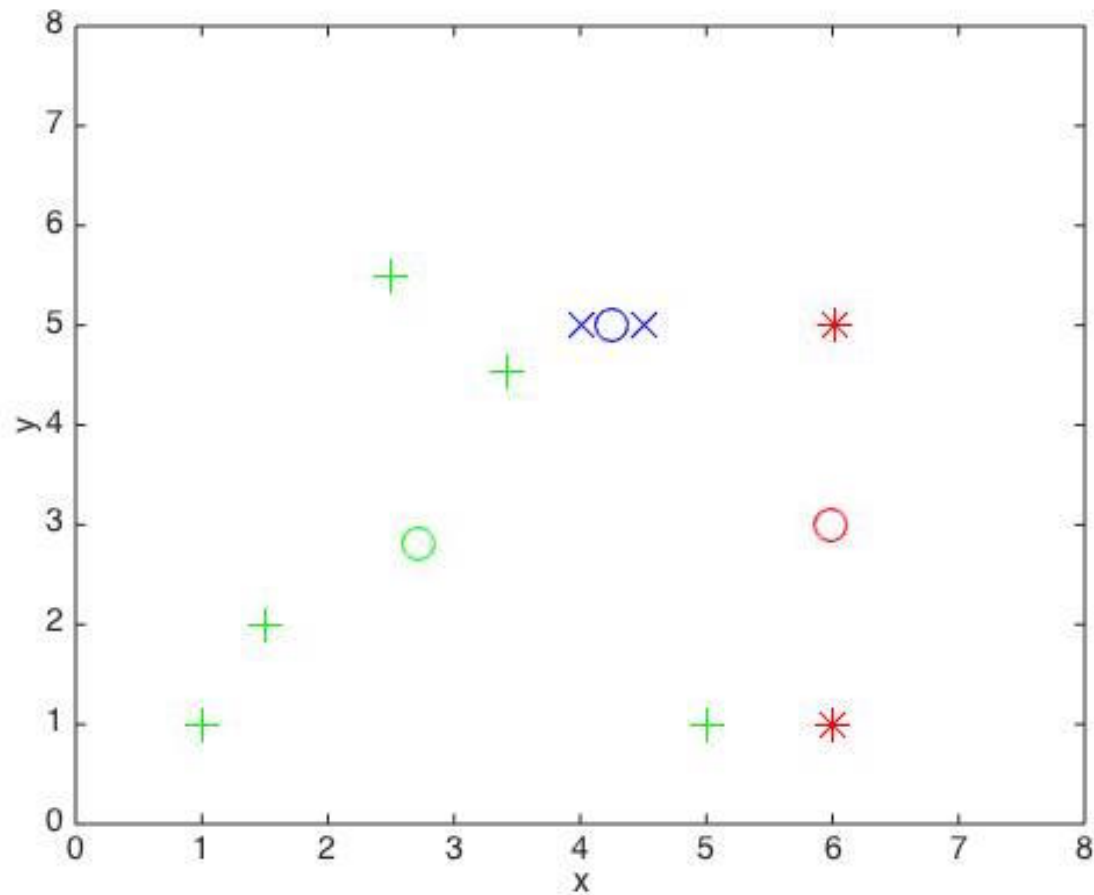




How to find new cluster centers if knowing cluster members?



New cluster centers can be the means of their corresponding members



# Can we do the two steps iteratively?

- Assign cluster membership based on cluster centers
- Find new cluster centers based on cluster membership

# Partitioning Algorithms: Basic Concept

- Partitioning method: Discovering the groupings in the data by optimizing a specific objective function and iteratively improving the quality of partitions
- K-partitioning method: Partitioning a dataset D of n objects into a set of K clusters so that an objective function is optimized
  - A typical objective function: Sum of Squared Errors (SSE), where  $c_k$  is the centroid or medoid of cluster  $C_k$

$$SSE = \sum_{i=1}^k \sum_{p \in C_i} (d(p, c_i))^2$$

- Problem definition: Given K, find a partition of K clusters that optimizes the chosen partitioning criterion
  - Global optimal: Needs to exhaustively enumerate all partitions
  - Heuristic methods (i.e., greedy algorithms): K-Means, K-Medians, K-Medoids, etc.

# K-Means: A typical cluster partitioning algorithm

**Algorithm:  $k$ -means.** The  $k$ -means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

**Input:**

- $k$ : the number of clusters,
- $D$ : a data set containing  $n$  objects.

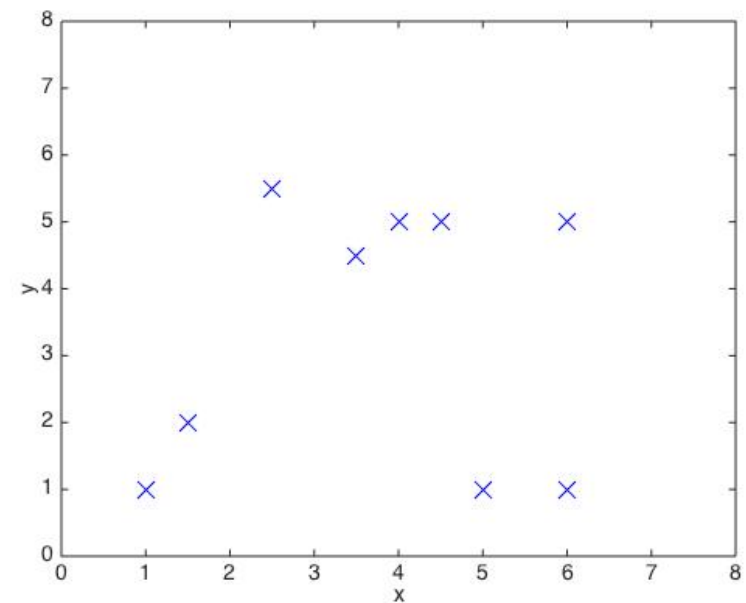
**Output:** A set of  $k$  clusters.

**Method:**

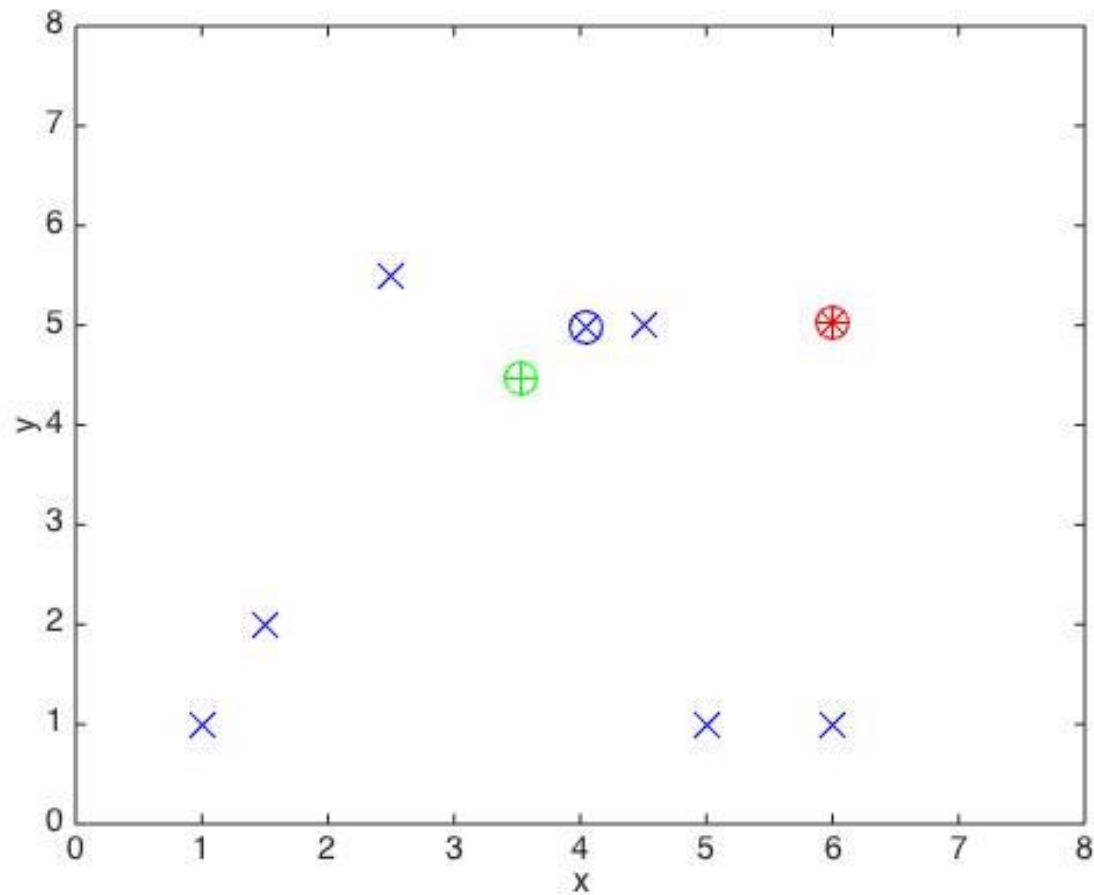
- (1) arbitrarily choose  $k$  objects from  $D$  as the initial cluster centers;
- (2) **repeat**
- (3)     (re)assign each object to the cluster to which the object is the most similar,  
          based on the mean value of the objects in the cluster;
- (4)     update the cluster means, that is, calculate the mean value of the objects for  
          each cluster;
- (5) **until** no change;

# K-Mean example: 2-D data points, K=3, and using Euclidean distance

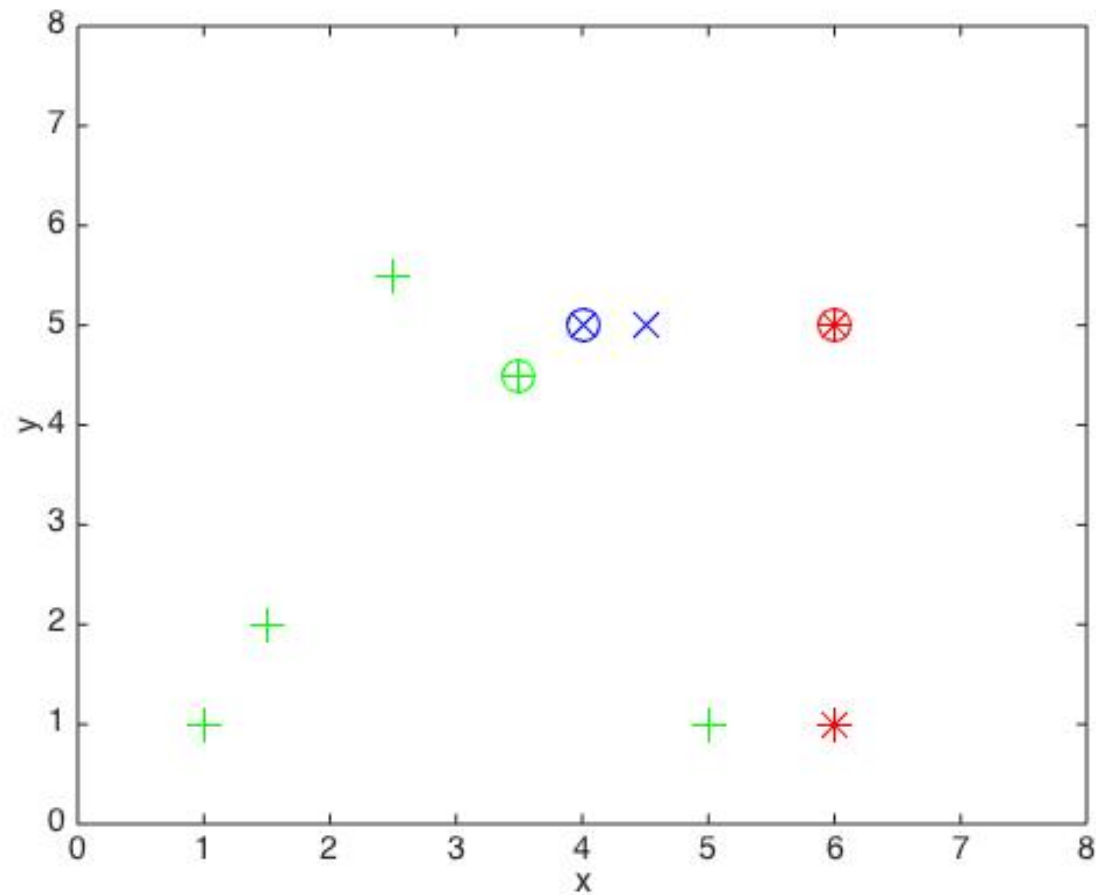
x	y
1	1
1.5	2
2.5	5
6	5
4	5
4.5	5
3.5	4.5
5	1
6	1



Initialization: Randomly choose 3 initial cluster centers

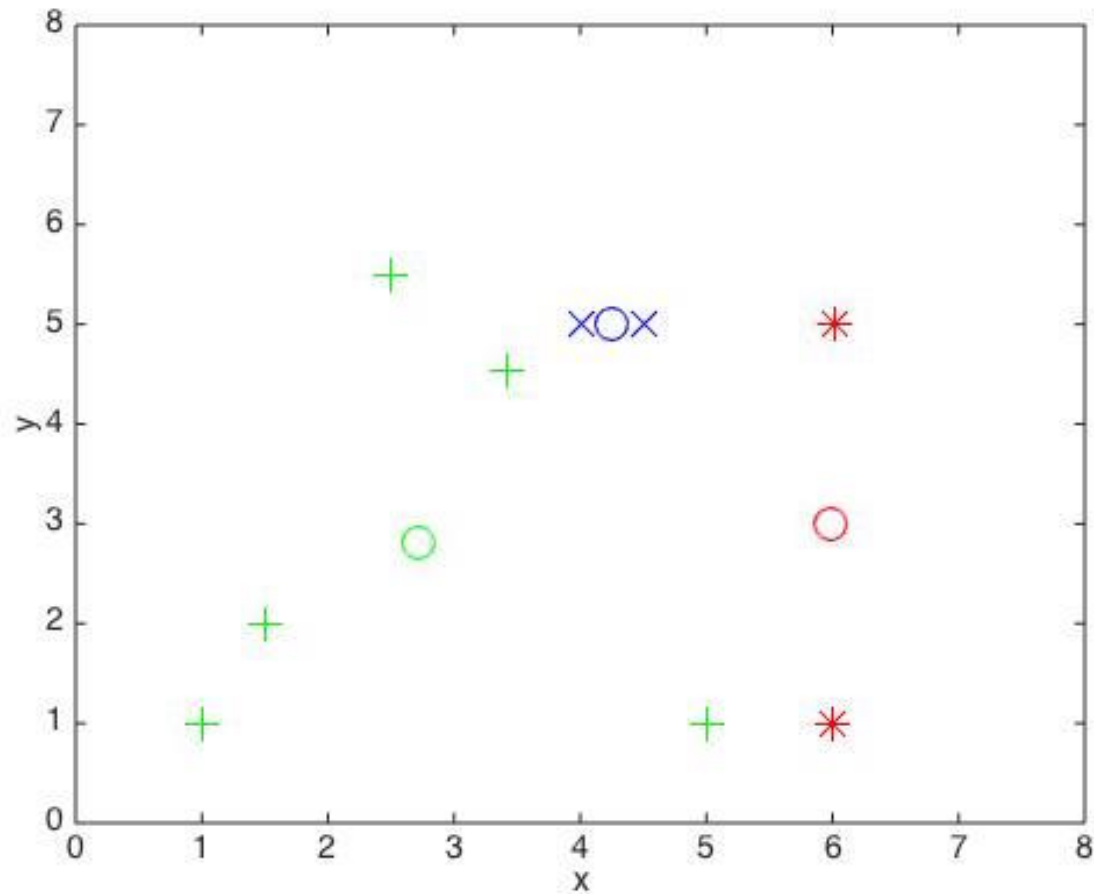


# Iter 1: Assigning points to closer cluster centers

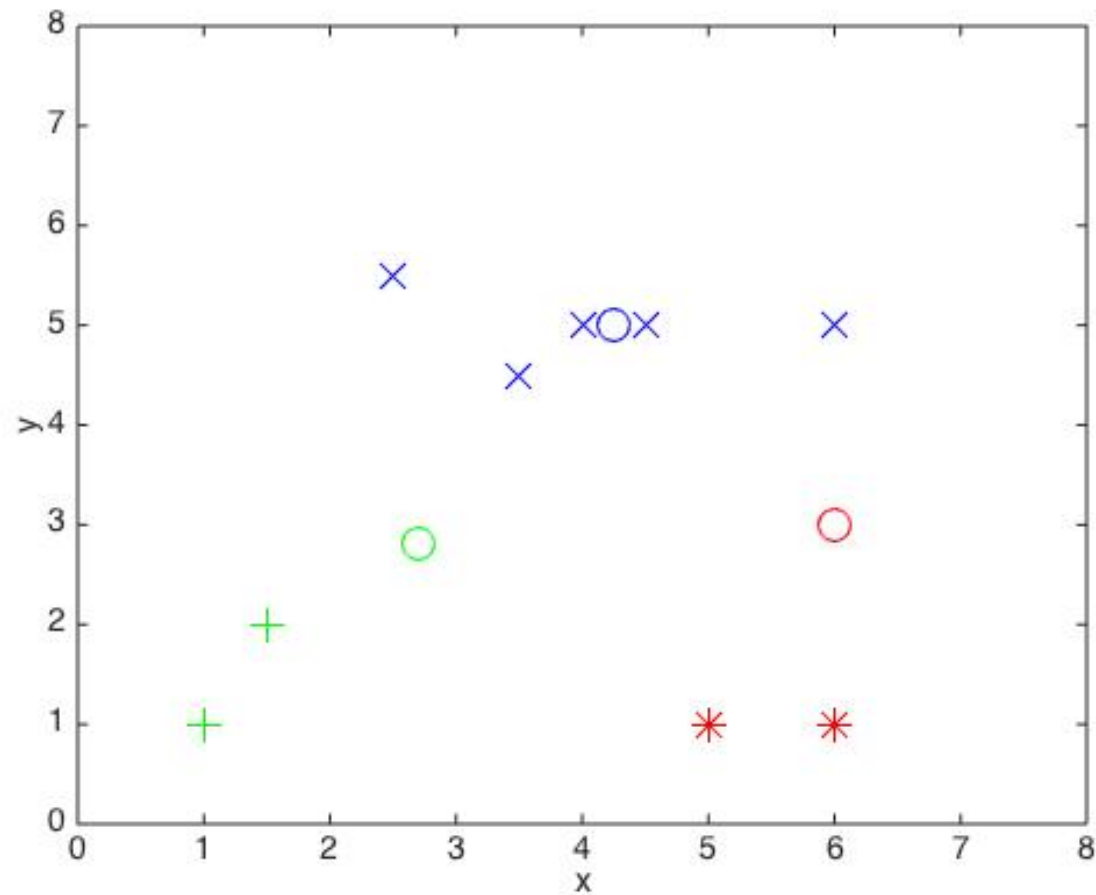




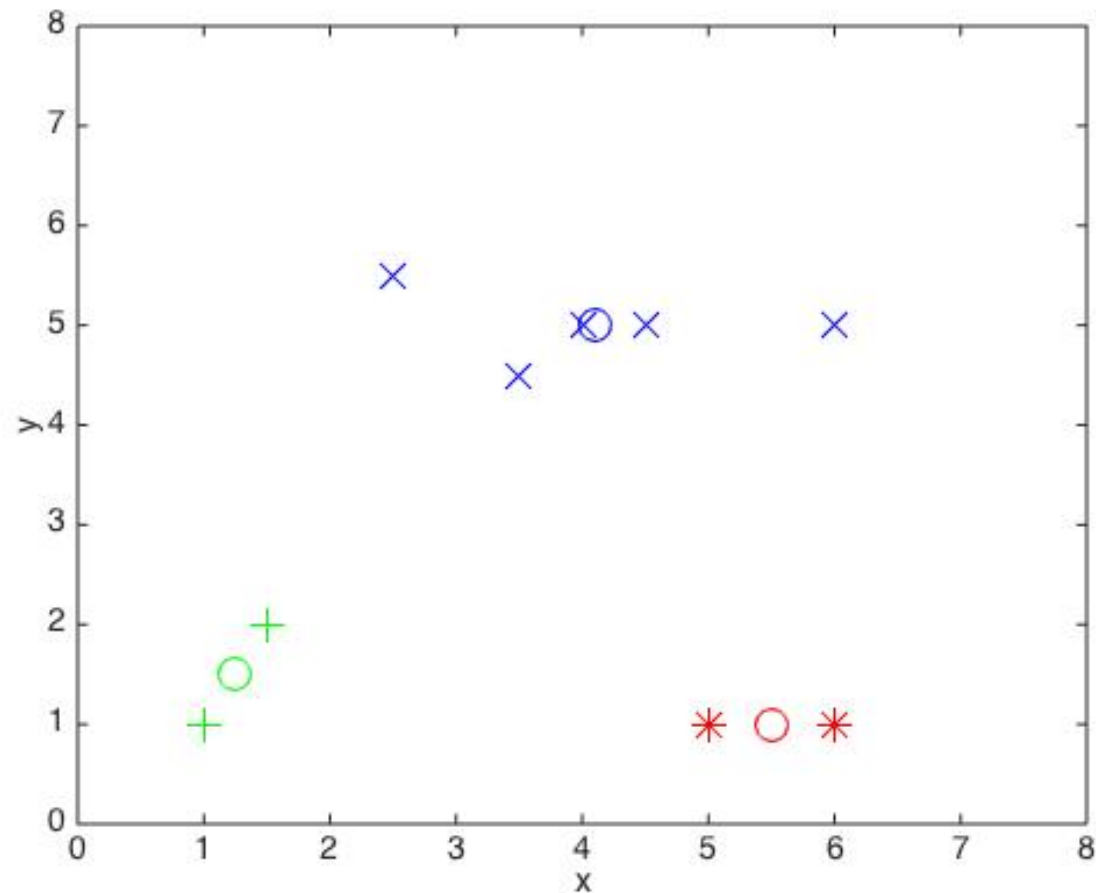
Iter 1: Find new cluster centers as the means of their corresponding members



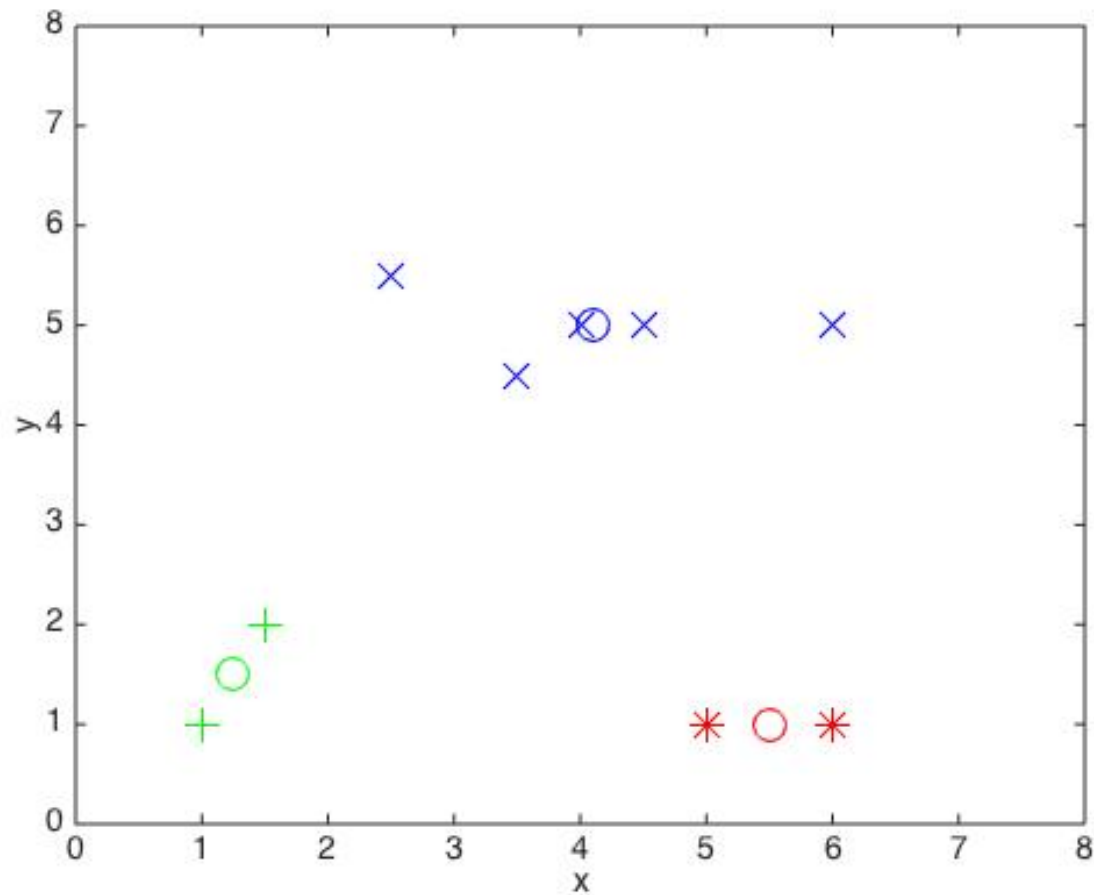
## Iter 2: Assigning points to closer cluster centers



Iter 2: Find new cluster centers as the means of their corresponding members



Iter 3: Stop because nothing changes

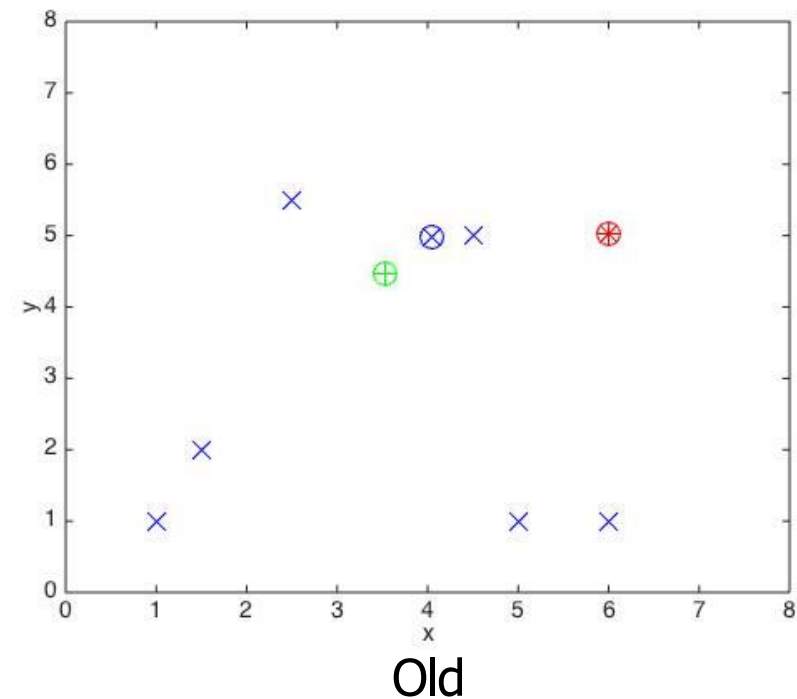
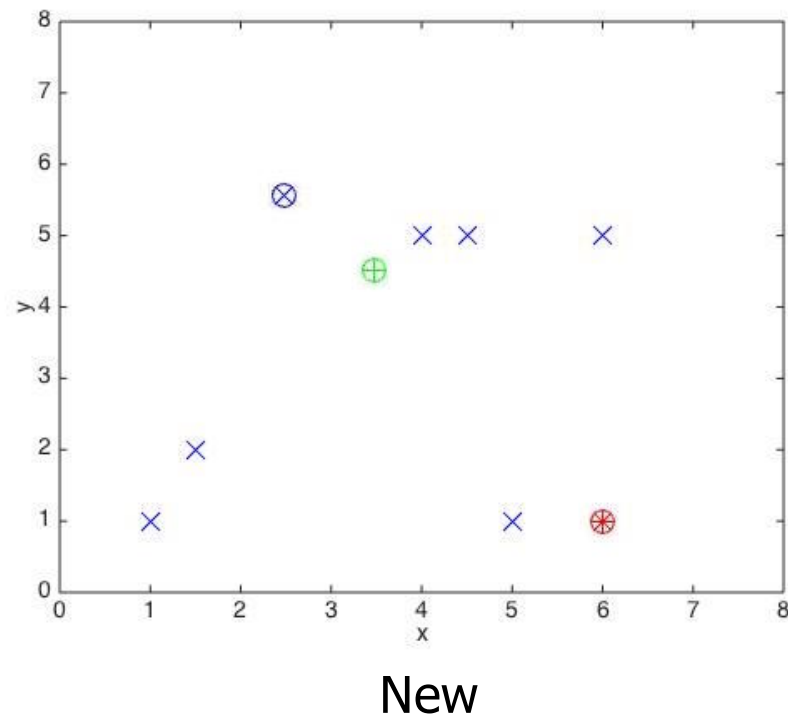


# Comments on K-Means algorithm

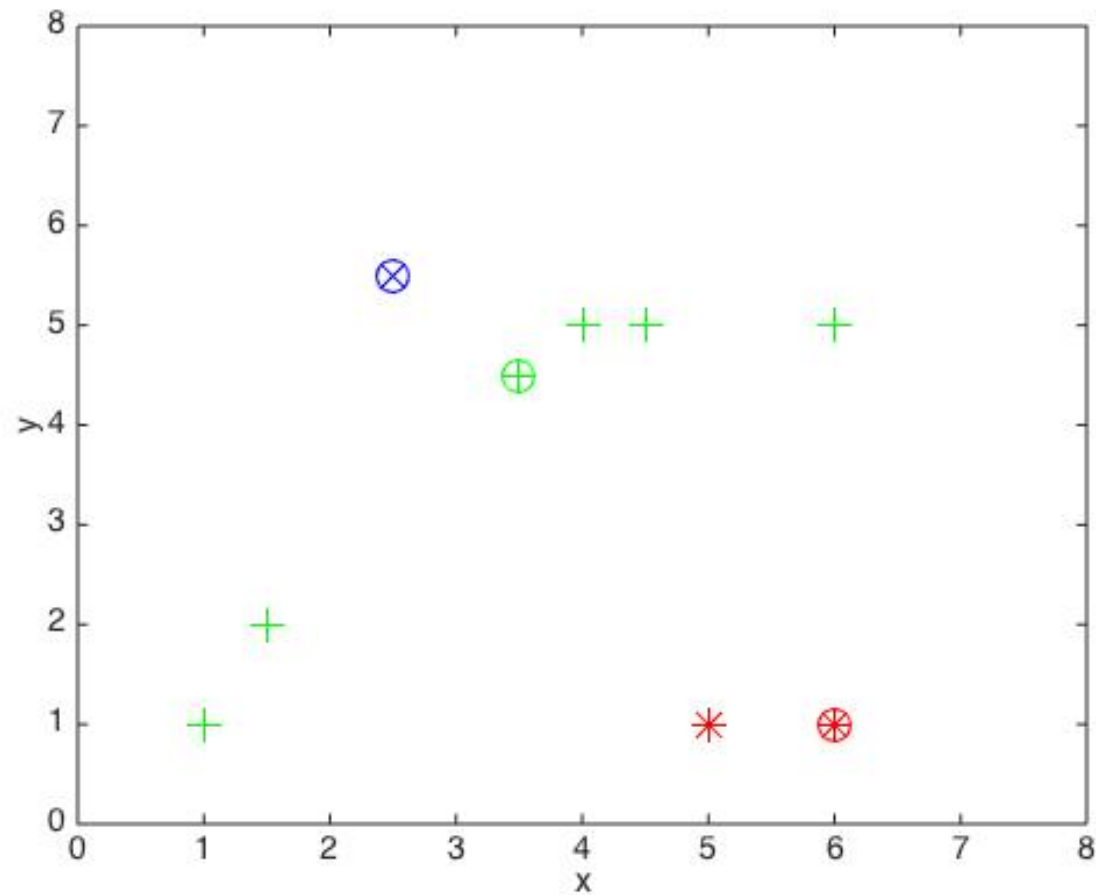
- Efficiency:  $O(tKn)$  where  $n$ : # of objects,  $K$ : # of clusters, and  $t$ : # of iterations
  - Normally,  $K, t \ll n$ ; thus, an efficient method
- Need to specify  $K$ , the number of clusters, in advance
  - There are ways to automatically determine the “best”  $K$
  - In practice, one often runs a range of values and selected the “best”  $K$  value
- Restricted usage:
  - Applicable only to objects in a continuous  $n$ -dimensional space
    - Using K-modes for categorical data
  - Not suitable to discover clusters with non-convex shapes
    - Using density-based clustering, kernel K-means, etc. instead
- Other than that: still have at least two issues

# Issue 1: K-means often terminates at a local optimal: initialization can be important to find high-quality clusters

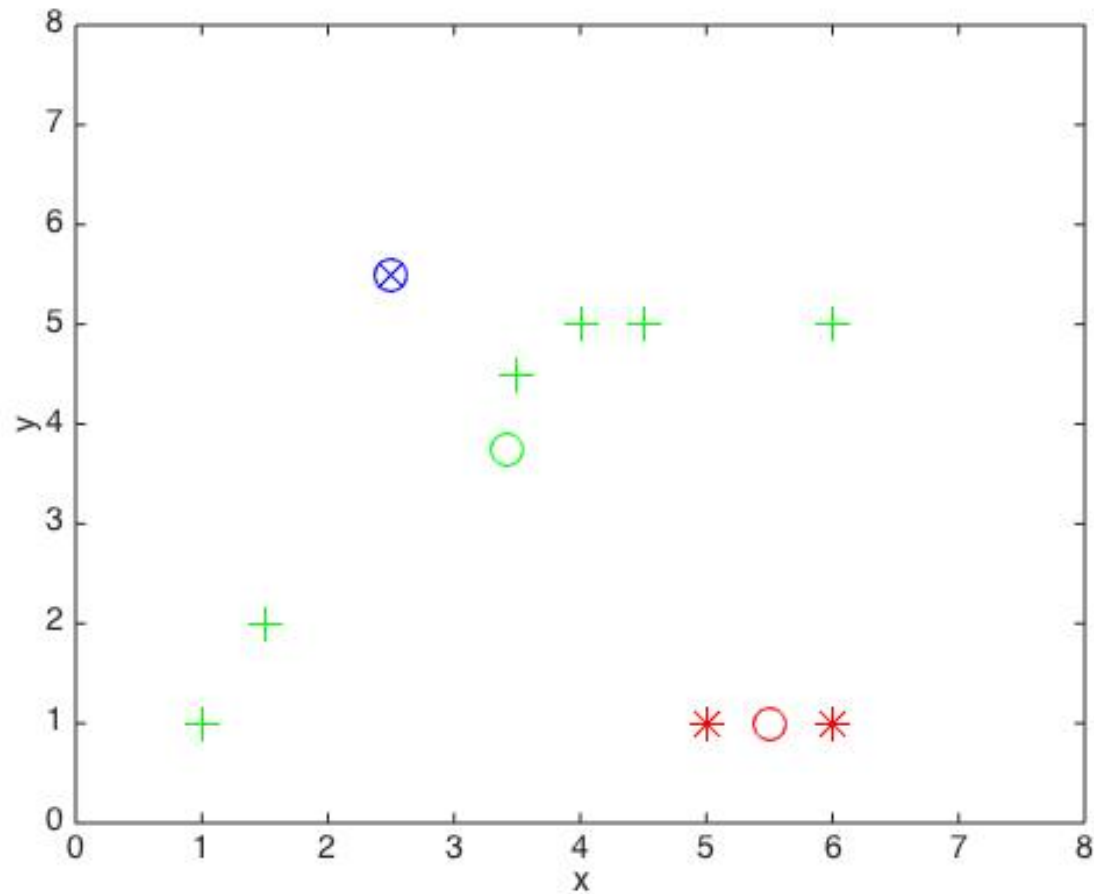
- Same data with the previous running example but with different initialization



# Iter 1: Assigning points to closer cluster centers

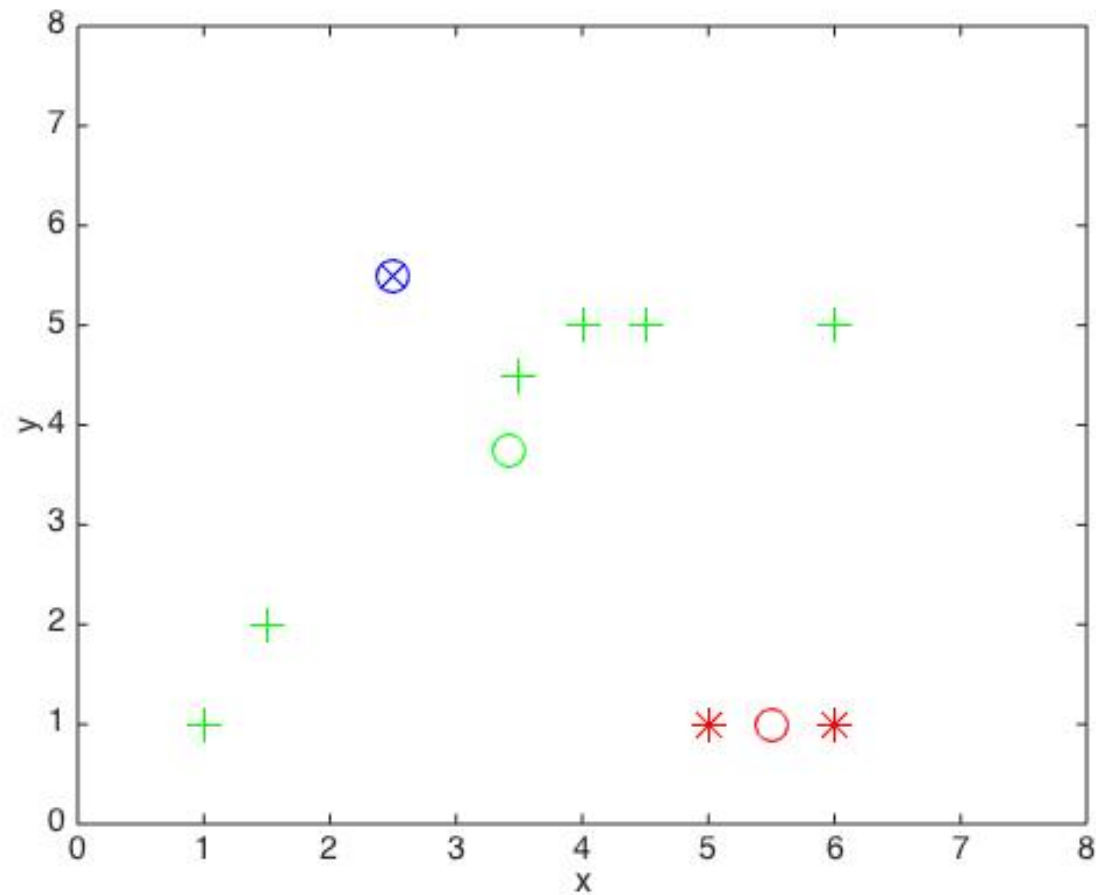


Iter 1: Find new cluster centers as the means of their corresponding members



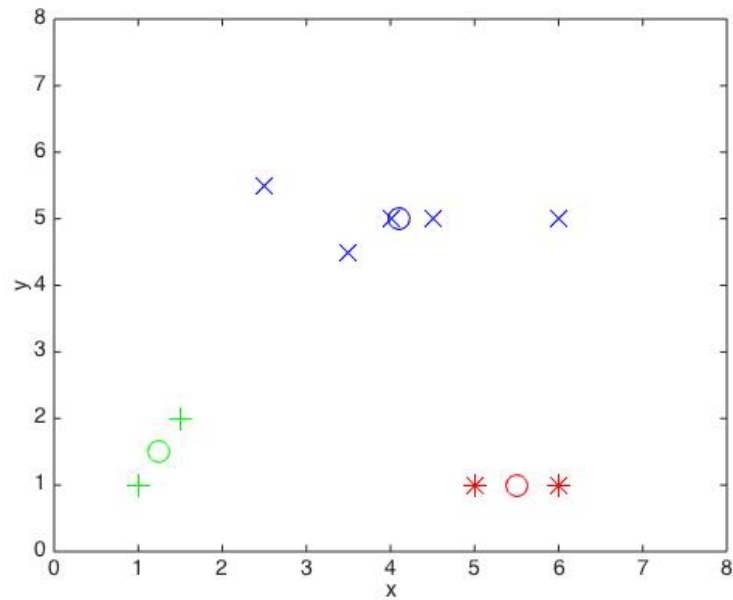


Iter 2: Stop because nothing changes

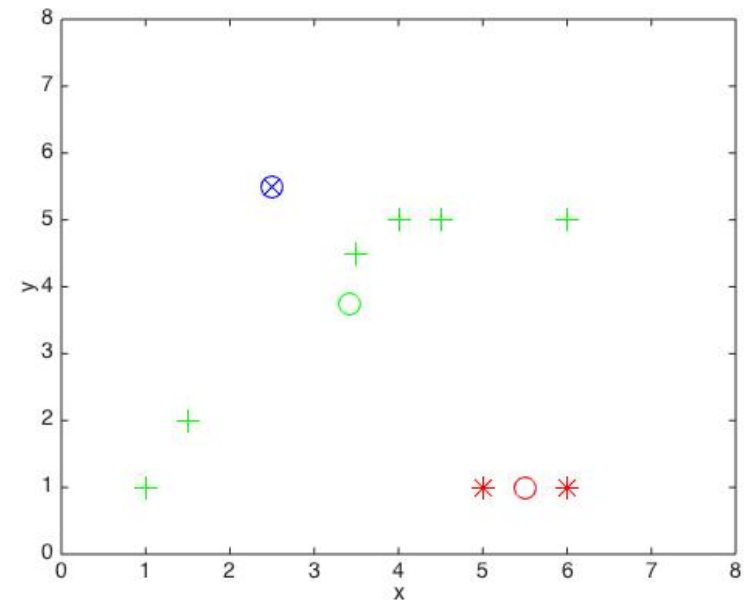


# Different results with different initialization

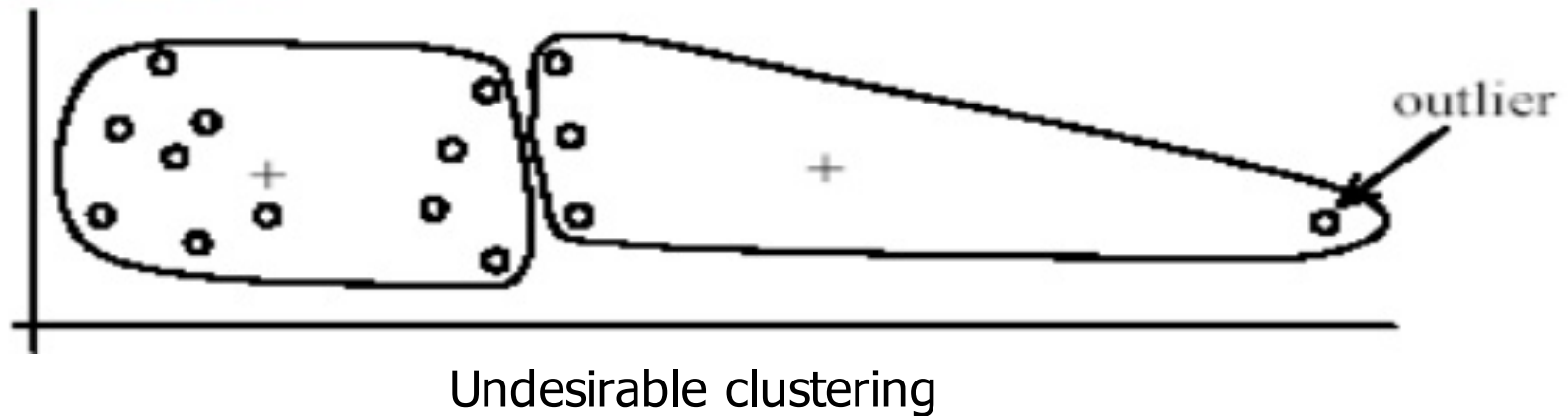
**Old**



**New**

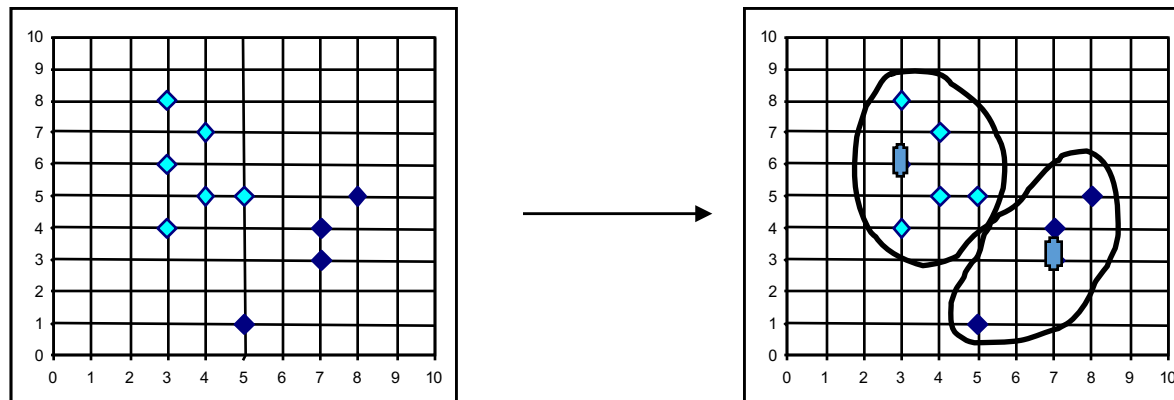


Issue 2. K-Means is sensitive to noisy data and outliers



# Handling outliers: From K-Means to K-Medoids

- The K-Means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster



## PAM (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987) A Typical K-Medoids Algorithm

**Algorithm:  $k$ -medoids.** PAM, a  $k$ -medoids algorithm for partitioning based on medoid or central objects.

**Input:**

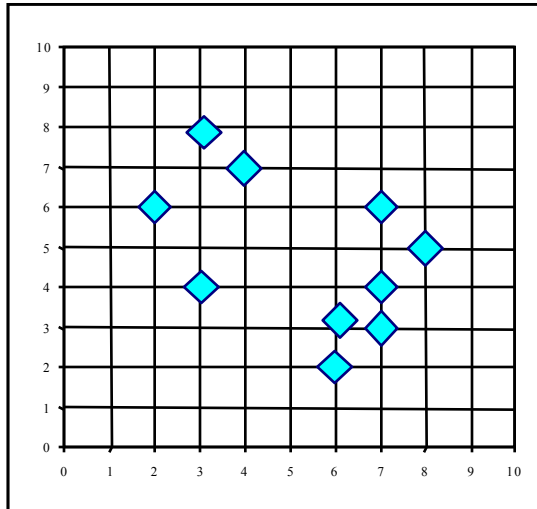
- $k$ : the number of clusters,
- $D$ : a data set containing  $n$  objects.

**Output:** A set of  $k$  clusters.

**Method:**

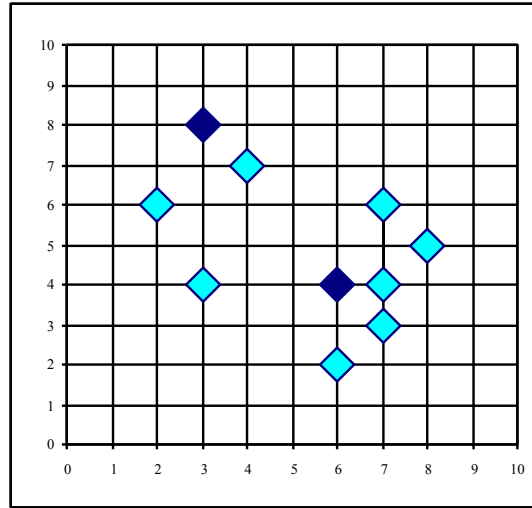
- (1) arbitrarily choose  $k$  objects in  $D$  as the initial representative objects or seeds;
- (2) **repeat**
- (3)     assign each remaining object to the cluster with the nearest representative object;
- (4)     randomly select a nonrepresentative object,  $o_{random}$ ;
- (5)     compute the total cost,  $S$ , of swapping representative object,  $o_j$ , with  $o_{random}$ ;
- (6)     **if**  $S < 0$  **then** swap  $o_j$  with  $o_{random}$  to form the new set of  $k$  representative objects;
- (7) **until** no change;

# PAM: A Typical K-Medoids Algorithm

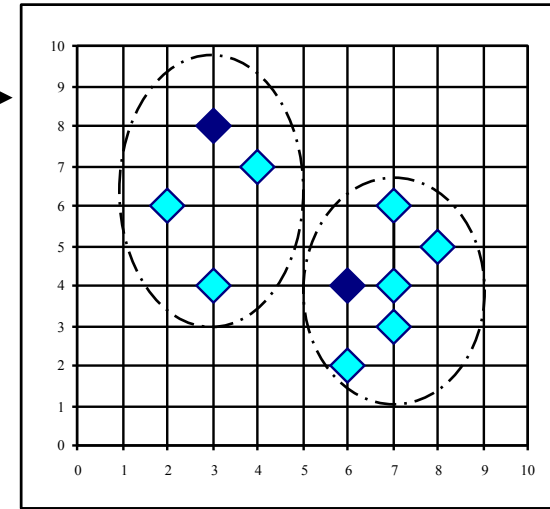


K=2

Arbitrary  
choose k  
object as  
initial  
medoids



Assign  
each  
remainin  
g object  
to  
nearest  
medoids



Total Cost = 20

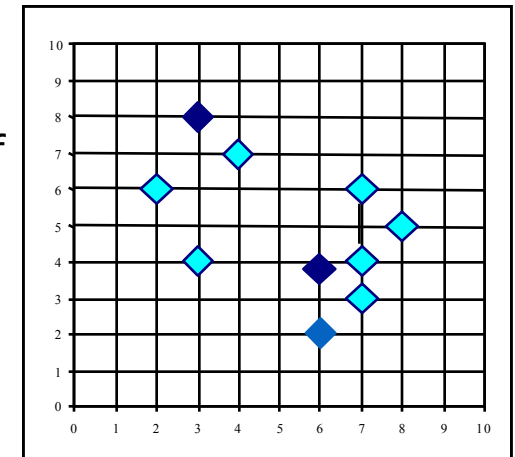
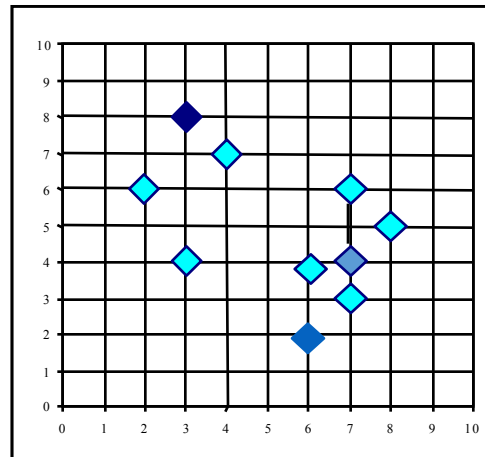
Total Cost = 26 > 20, so no swapping

Randomly select a  
nonmedoid object,  $O_{\text{random}}$

**Do loop  
Until no  
change**

Swapping  $O$   
and  $O_{\text{random}}$   
If quality is  
improved.


Compute  
total cost of  
swapping



# Efficiency issue of K-Medoids

- PAM does not scale well for large dataset
- Efficiency:
  - PAM:  $O(k(n-k)^2)$
  - Vs. K-Means  $O(tkn)$  where normally,  $k, t \ll n$
- Efficiency improvement on PAM
  - CLARA (Kaufmann & Rousseeuw, 1990): PAM on samples
  - CLARANS (Ng & Han, 1994): Randomized re-sampling

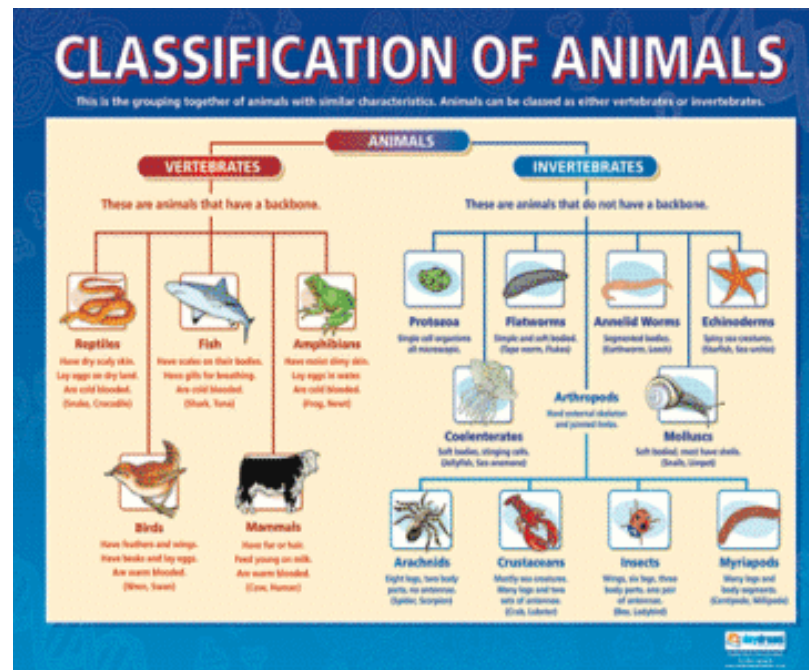
# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods 
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary



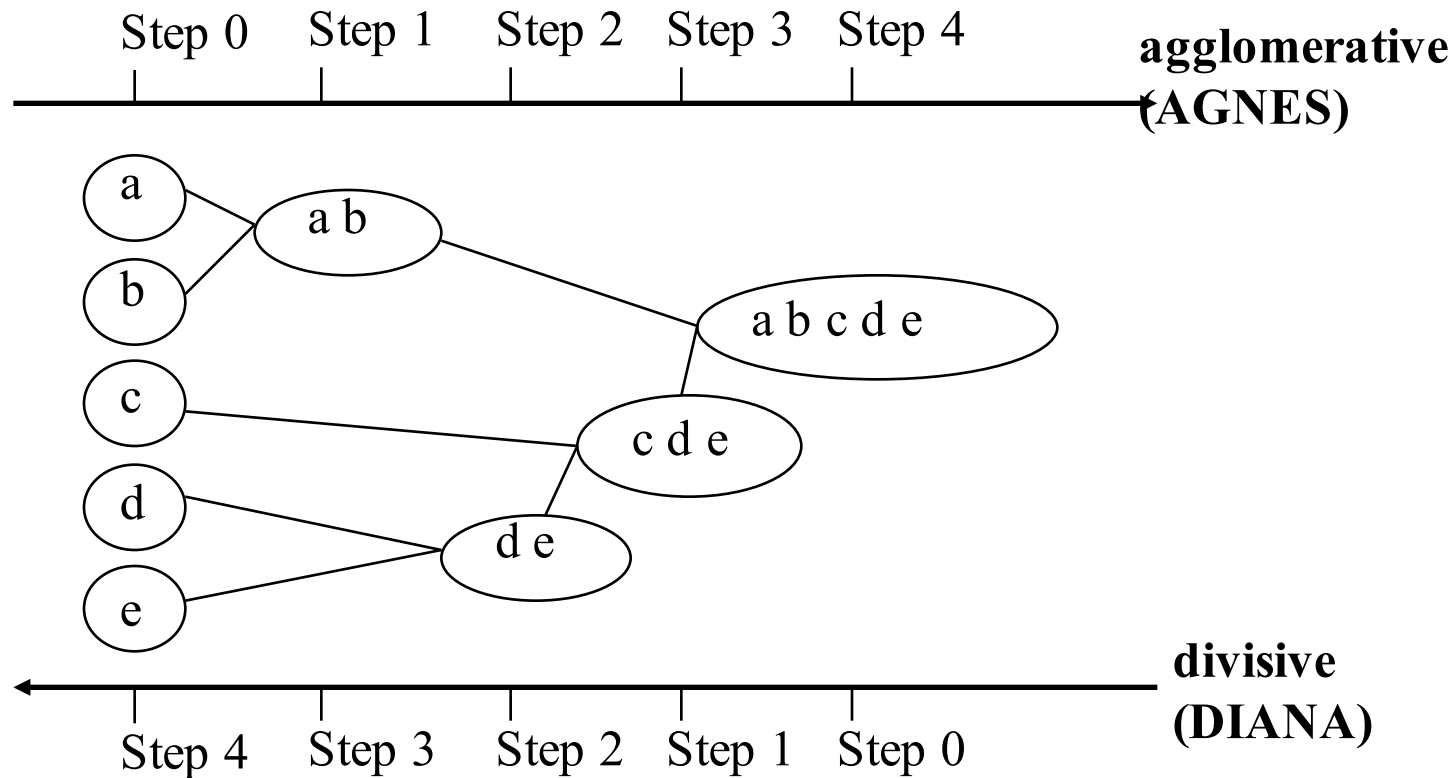
# Hierarchical clustering arises when clusters are nested

- What if the CEO in our example wants to cluster his staffs into a hierarchy instead of groups?
- What if scientists want to cluster species into a taxonomy instead of groups?
- Partitioning algorithms cannot help!



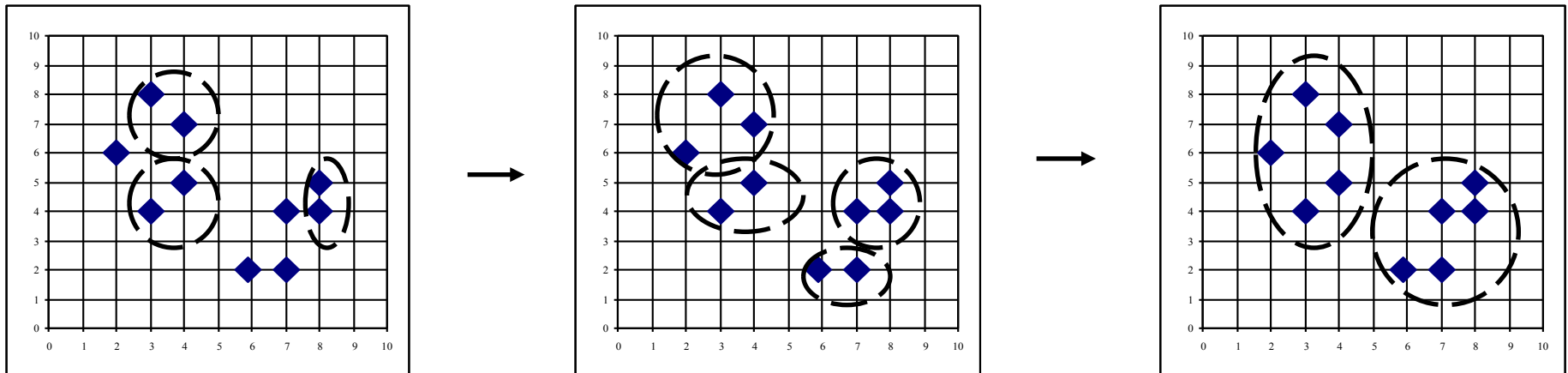
# Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters  $k$  as an input, but needs a termination condition



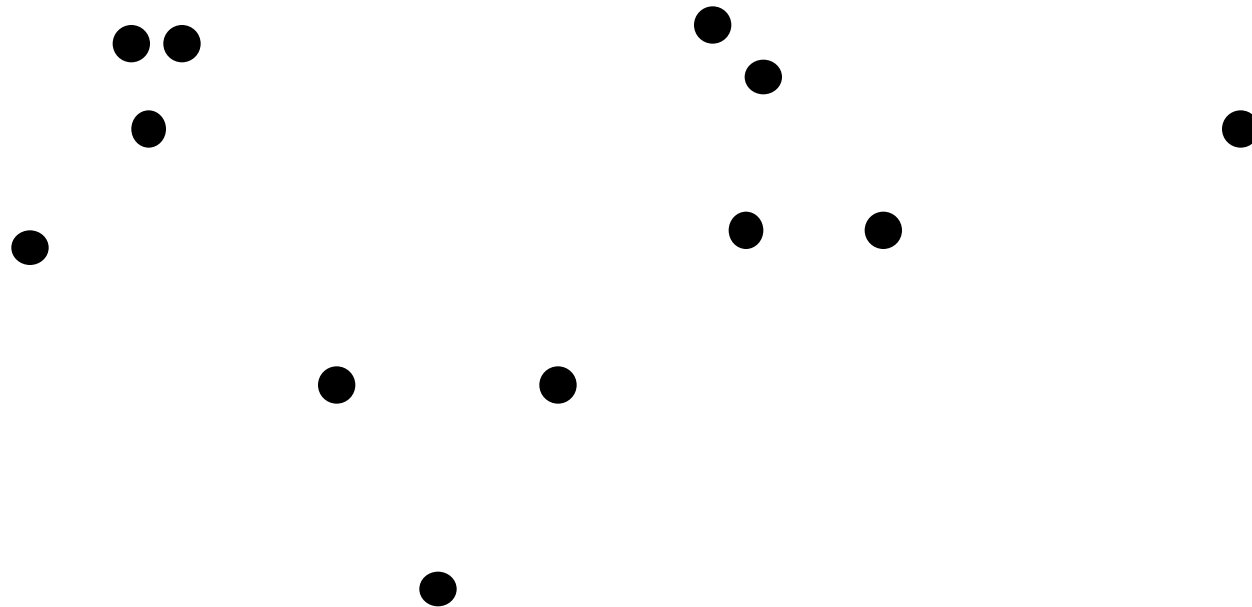
# AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the single-link method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



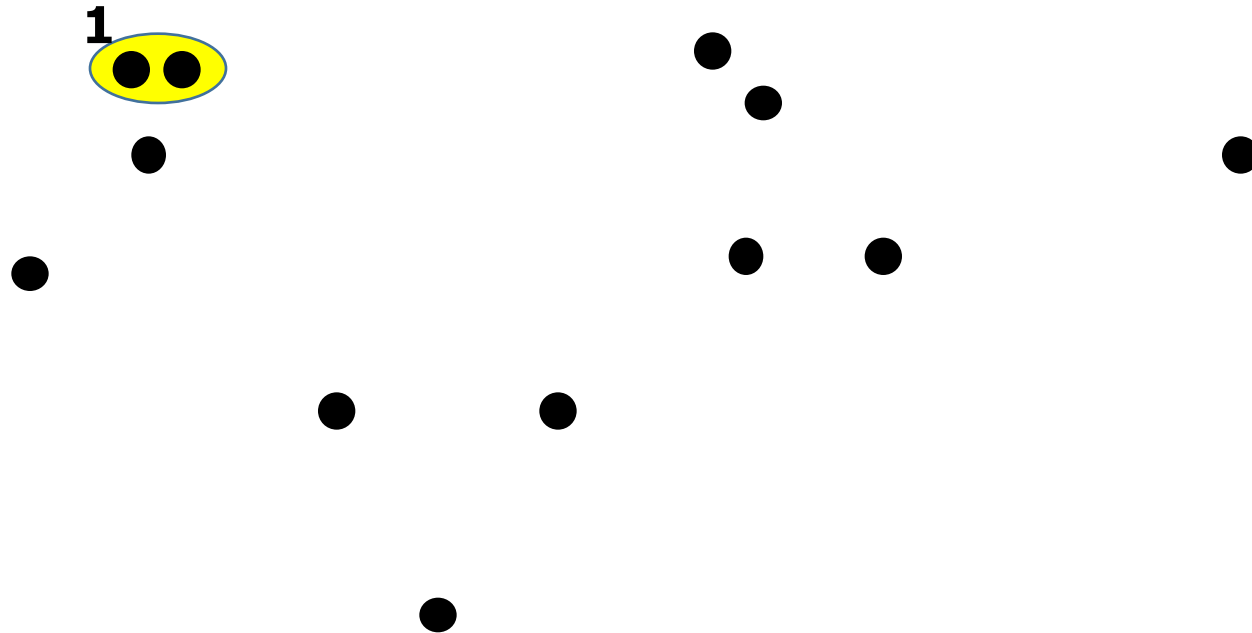
# AGNES (Agglomerative Nesting) in action

- Agglomerative (Bottom up)



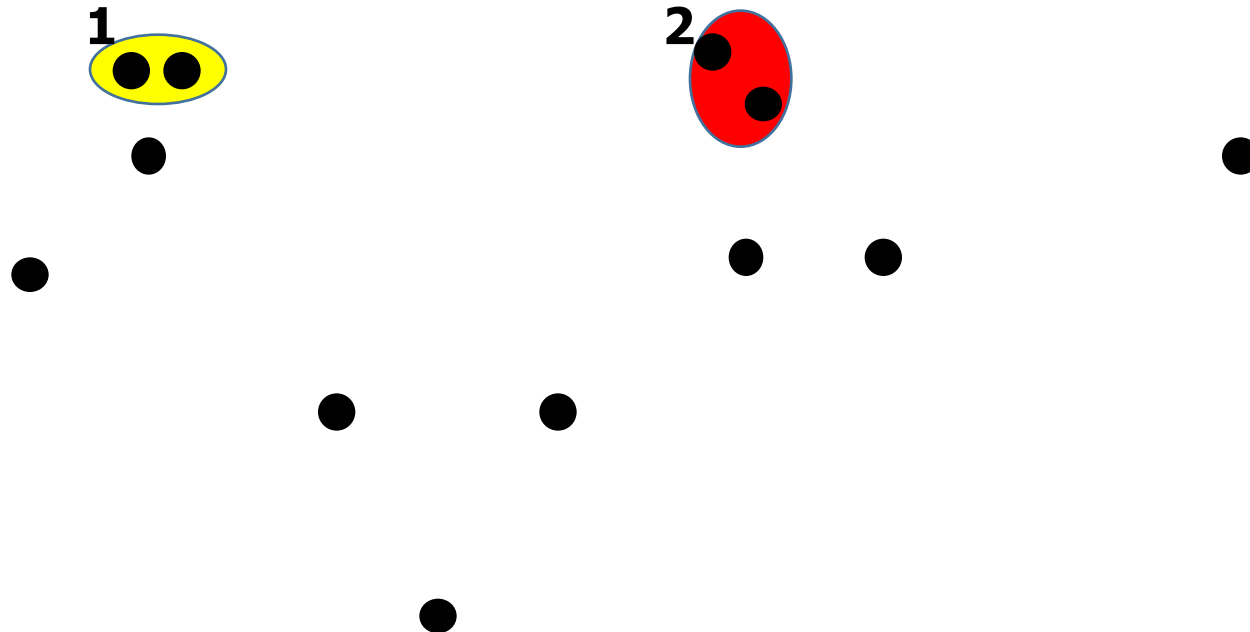
# AGNES (Agglomerative Nesting) in action

- Agglomerative (Bottom up)
- 1<sup>st</sup> iteration



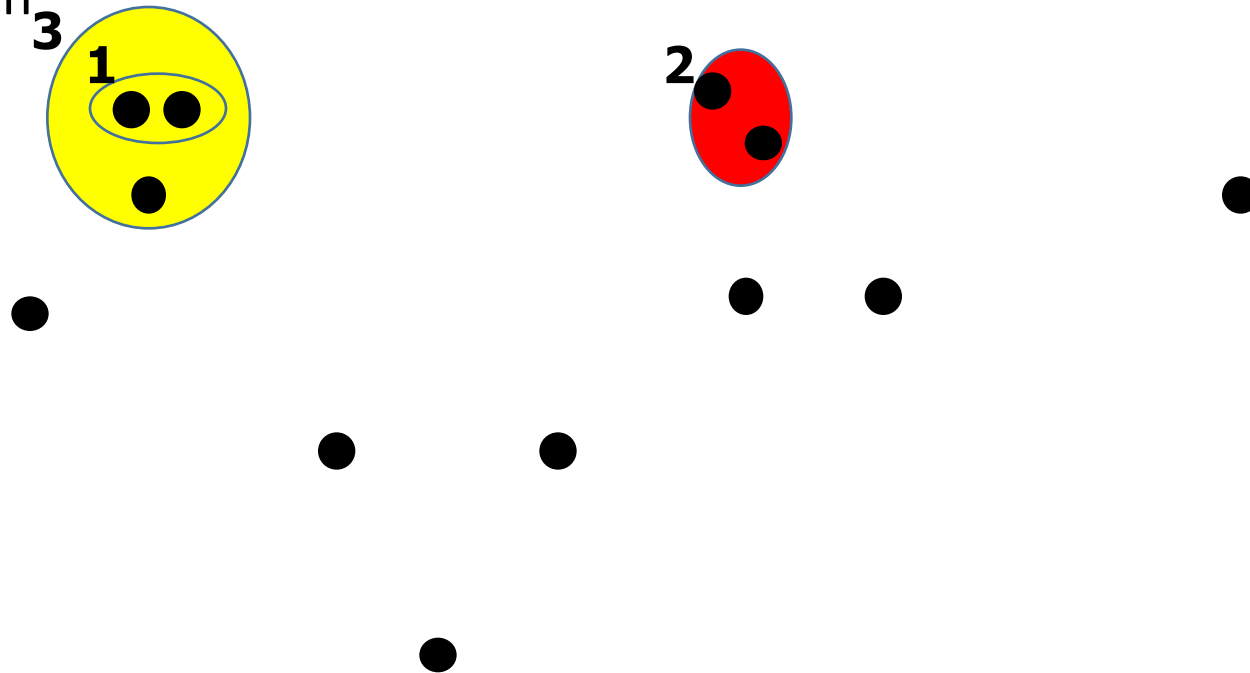
# AGNES (Agglomerative Nesting) in action

- Agglomerative (Bottom up)
- 2<sup>nd</sup> iteration



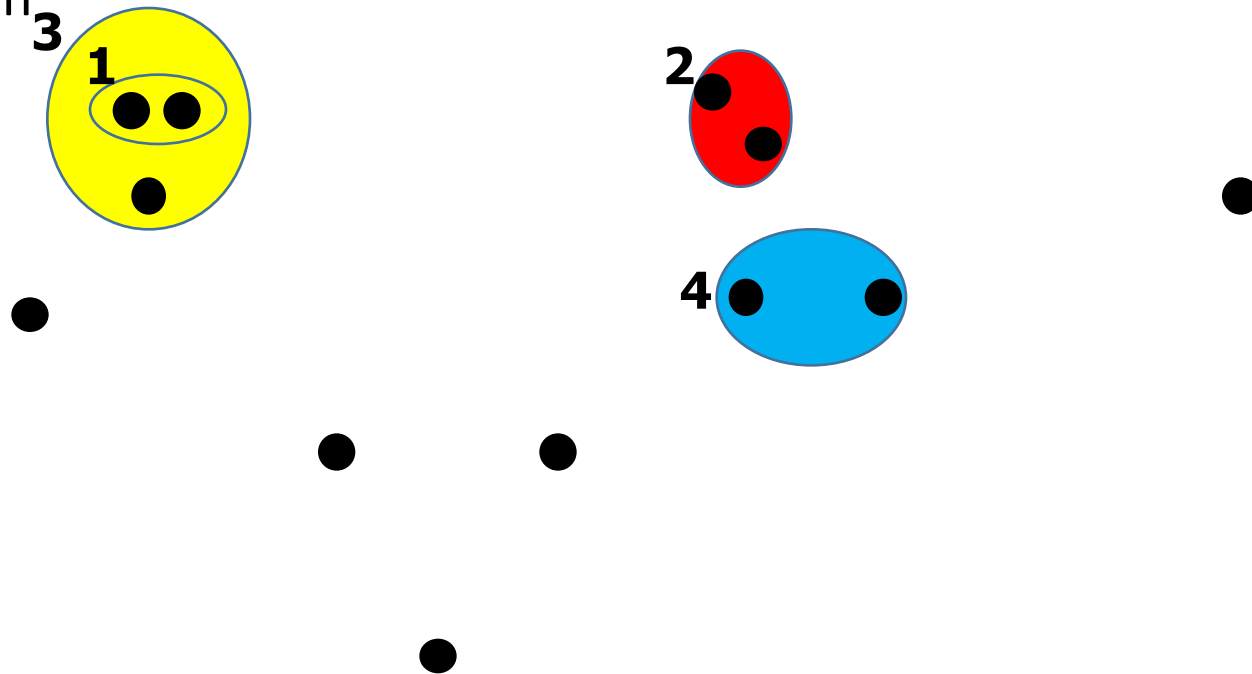
# AGNES (Agglomerative Nesting) in action

- Agglomerative (Bottom up)
- 3<sup>rd</sup> iteration



# AGNES (Agglomerative Nesting) in action

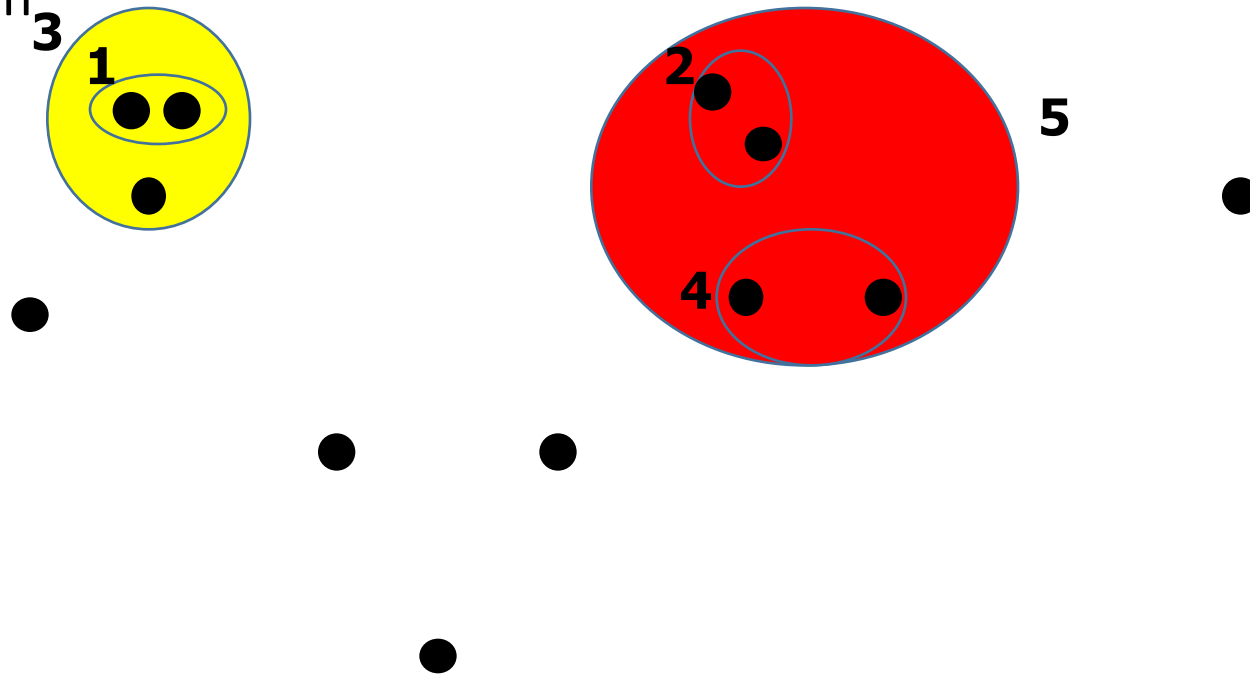
- Agglomerative (Bottom up)
- 4<sup>th</sup> iteration





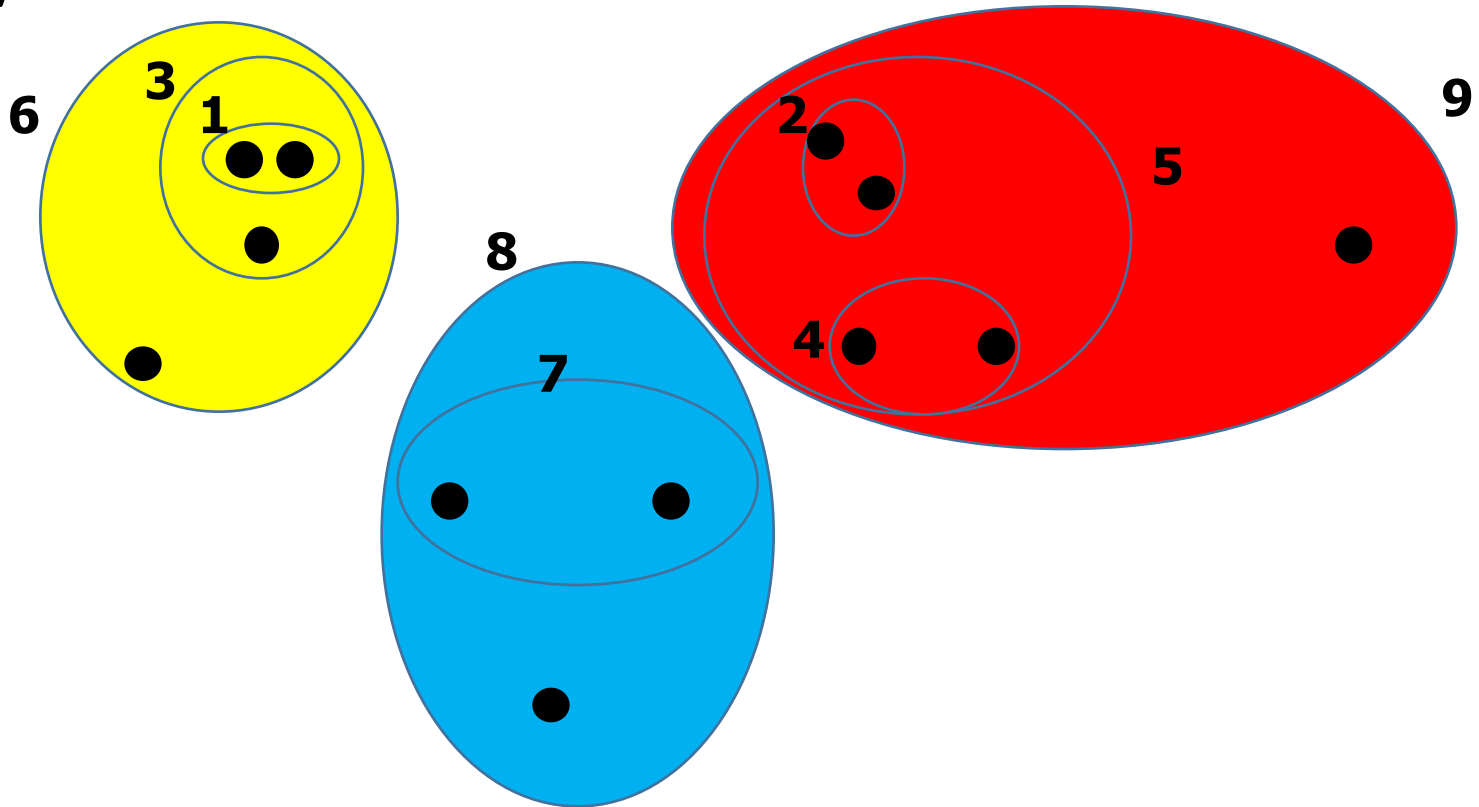
# AGNES (Agglomerative Nesting) in action

- Agglomerative (Bottom up)
- 5<sup>th</sup> iteration



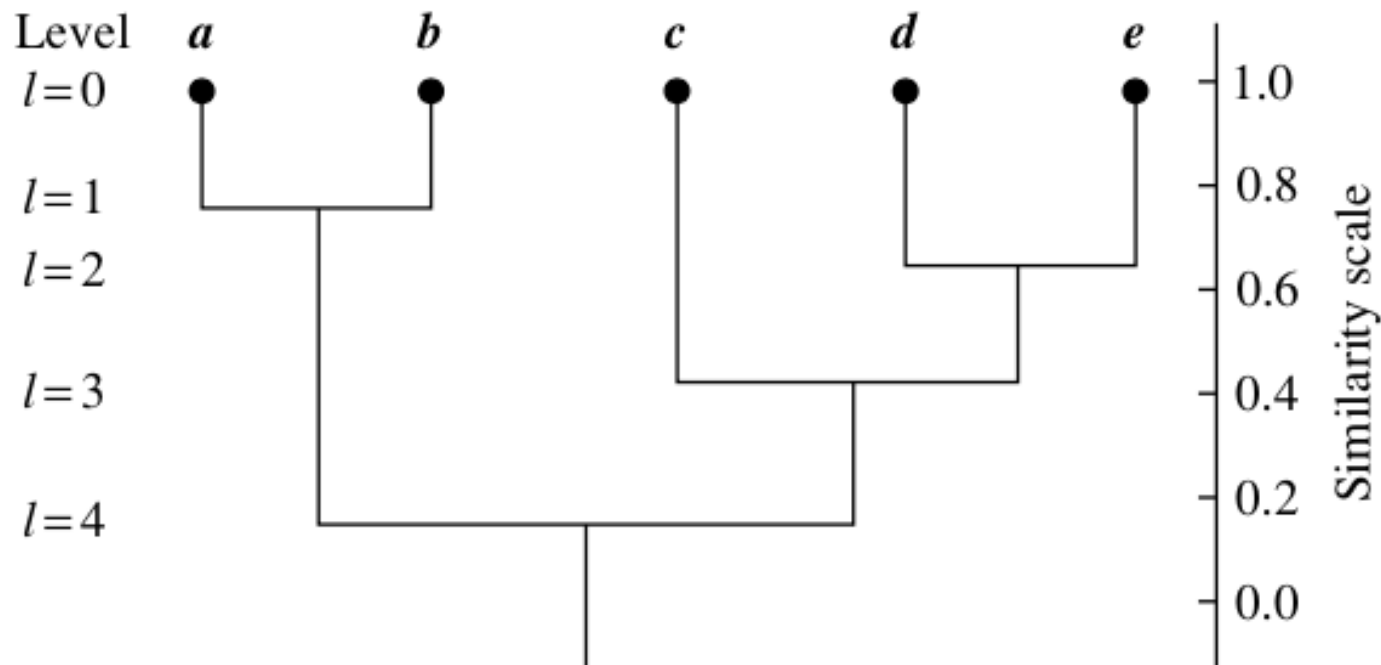
# AGNES (Agglomerative Nesting) in action

- Agglomerative (Bottom up)
- Finally k clusters left



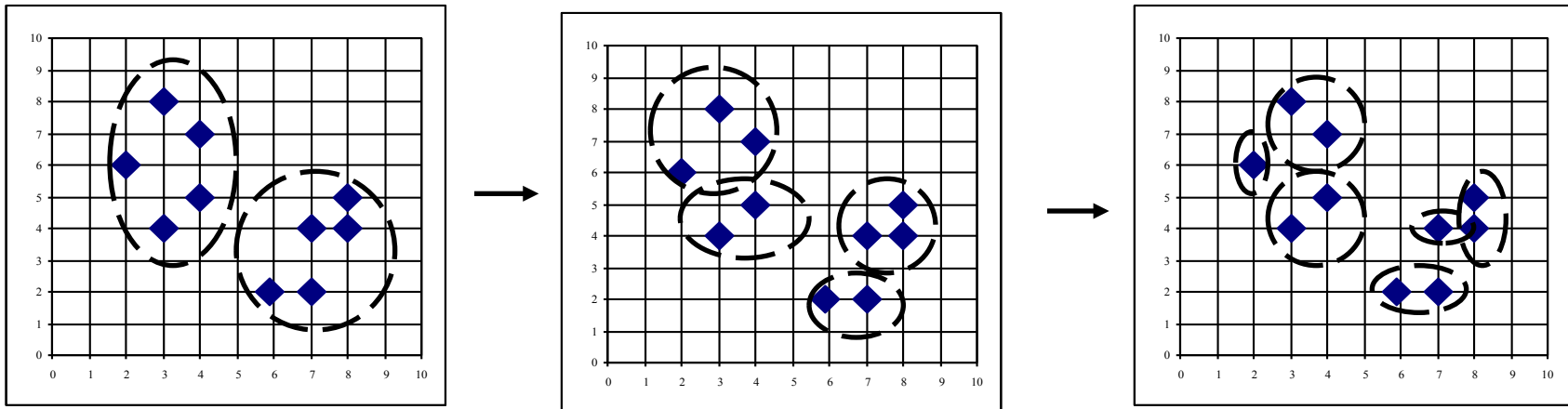
# Dendrogram: Shows How Clusters are Merged

- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

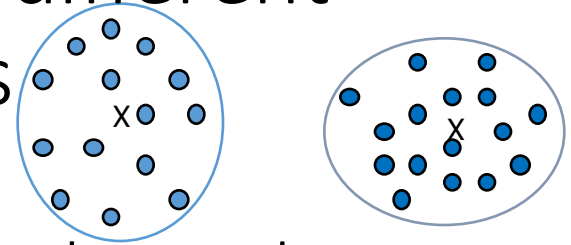


# DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



# Agglomerative clustering varies on different similarity measures among clusters

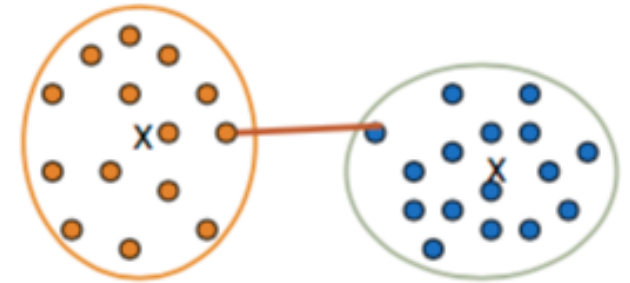


- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e.,  $\text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e.,  $\text{dist}(K_i, K_j) = \max(t_{ip}, t_{jq})$
- **Average:** avg distance between an element in one cluster and an element in the other, i.e.,  $\text{dist}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})$
- **Centroid:** distance between the centroids of two clusters, i.e.,  $\text{dist}(K_i, K_j) = \text{dist}(C_i, C_j)$
- **Medoid:** distance between the medoids of two clusters, i.e.,  $\text{dist}(K_i, K_j) = \text{dist}(M_i, M_j)$ 
  - Medoid: a chosen, centrally located object in the cluster

# Single Link vs. Complete Link in Hierarchical Clustering

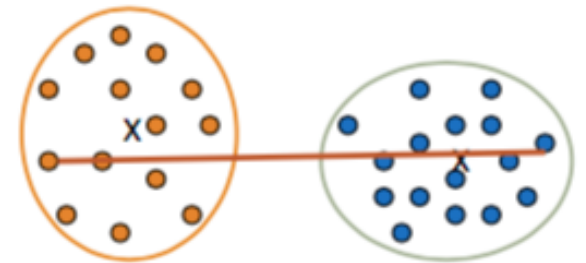
- Single link (nearest neighbor)

- The similarity between two clusters is the similarity between
- their most similar (nearest neighbor) members
- Local similarity-based: Emphasizing more on close regions, ignoring the overall structure of the cluster
- Capable of clustering non-elliptical shaped group of objects
- Sensitive to noise and outliers



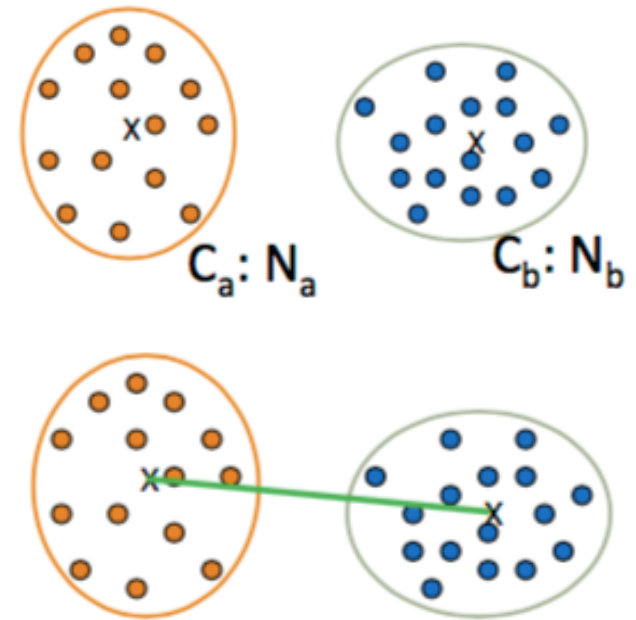
- Complete link (diameter)

- The similarity between two clusters is the similarity between their most dissimilar members
- Merge two clusters to form one with the smallest diameter
- Nonlocal in behavior, obtaining compact shaped clusters
- Sensitive to outliers



# Agglomerative Clustering: Average vs. Centroid Links

- Agglomerative clustering with average link
  - Average link: The average distance between an element in one cluster and an element in the other (i.e., all pairs in two clusters)
  - Expensive to compute
- Agglomerative clustering with centroid link
  - Centroid link: The distance between the centroids of two clusters




# Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Do not scale well: time complexity of at least  $O(n^2)$ , where  $n$  is the number of total objects
- Integration of hierarchical & distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

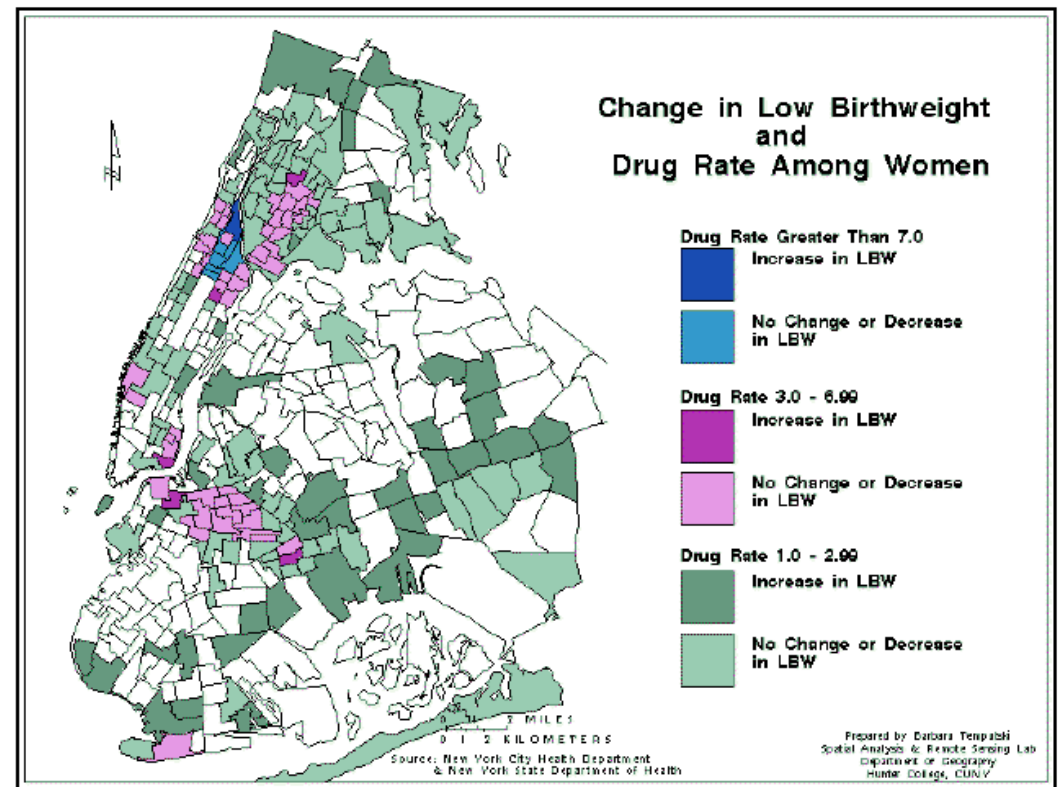


# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods 
- Grid-Based Methods
- Evaluation of Clustering
- Summary

# Density-based methods arises when clustering large spatial databases

- **R1. Minimal requirements of domain knowledge** to determine the input parameters, because appropriate values are often not known in advance when dealing with large databases.
- **R2. Discovery of clusters with arbitrary shape**, because the shape of clusters in spatial databases may be spherical, drawn-out, linear, elongated etc.
- **R3. Good efficiency on large databases**, i.e. on databases of significantly more than just a few thousand objects.



# Density-Based Clustering Methods

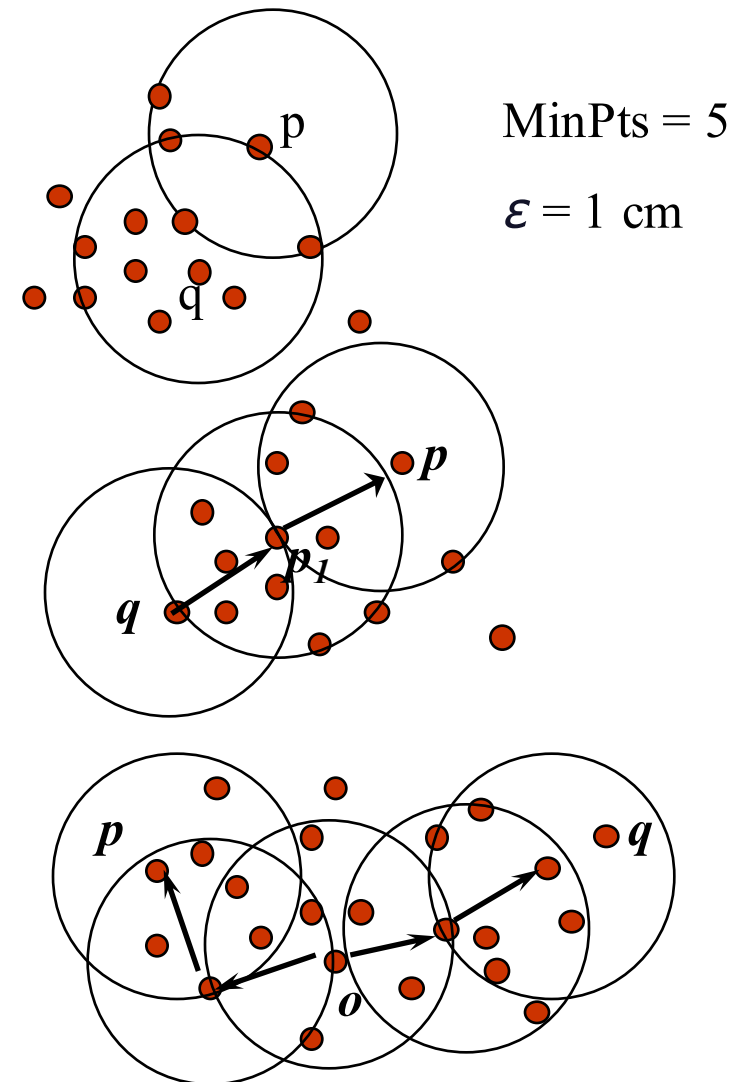
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape: dense area → R2
  - Discover noise: if points do not belong to a dense area → R2
  - Need density parameters as termination condition → R1
  - Require one scan → R3
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

# Density-Based Clustering: Density parameters

- Two parameters:
  - $\epsilon$ : Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- *Neighboring points*  $N_\epsilon(q) = \{p \in D \mid \text{dist}(p, q) \leq \epsilon\}$ 
  - Note:  $q \in N_\epsilon(q)$

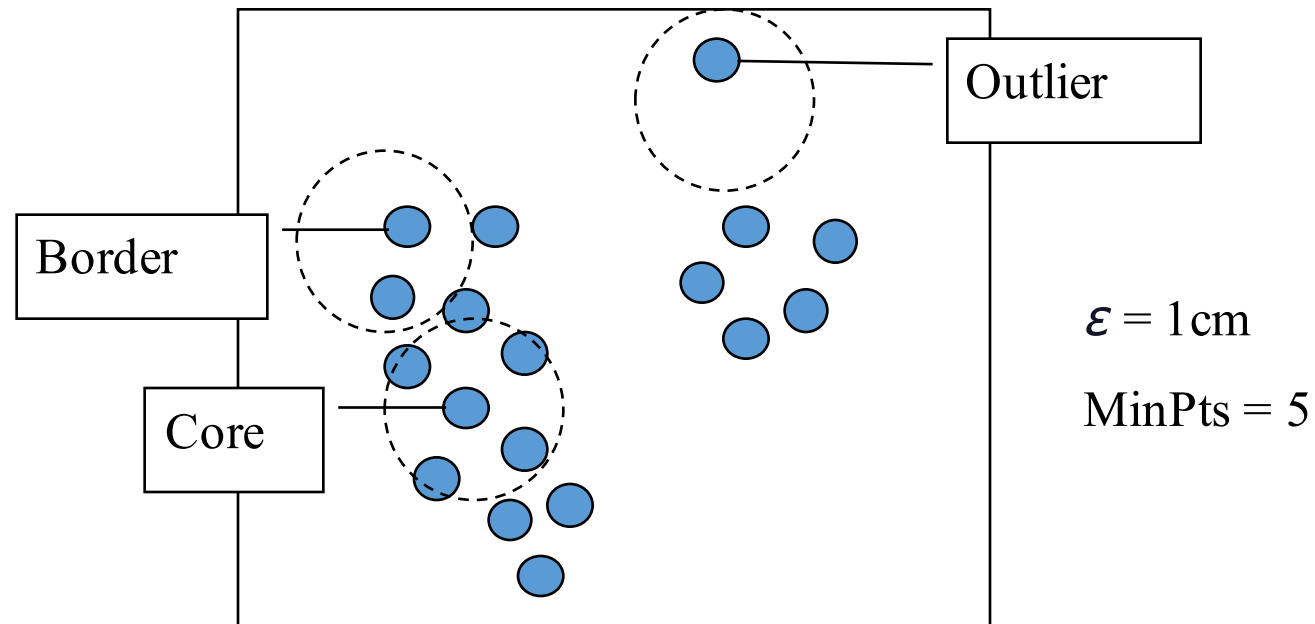
# Density-Based Clustering: Density measures

- **Directly density-reachable:** A point  $p$  is directly density-reachable from a point  $q$  w.r.t.  $\epsilon$ , MinPts if
  - $p$  belongs to  $N_\epsilon(q)$
  - core point condition:  $|N_\epsilon(q)| \geq \text{MinPts}$
- **Density-reachable:**
  - A point  $p$  is density-reachable from a point  $q$  w.r.t.  $\epsilon$ , MinPts if there is a chain of points  $p_1, \dots, p_n$ ,  $p_1 = q$ ,  $p_n = p$  such that  $p_{i+1}$  is directly density-reachable from  $p_i$
- **Density-connected**
  - A point  $p$  is density-connected to a point  $q$  w.r.t.  $\epsilon$ , MinPts if there is a point  $o$  such that both  $p$  and  $q$  are density-reachable from  $o$  w.r.t.  $\epsilon$  and MinPts
  - Two density-connected points should belong to the same cluster

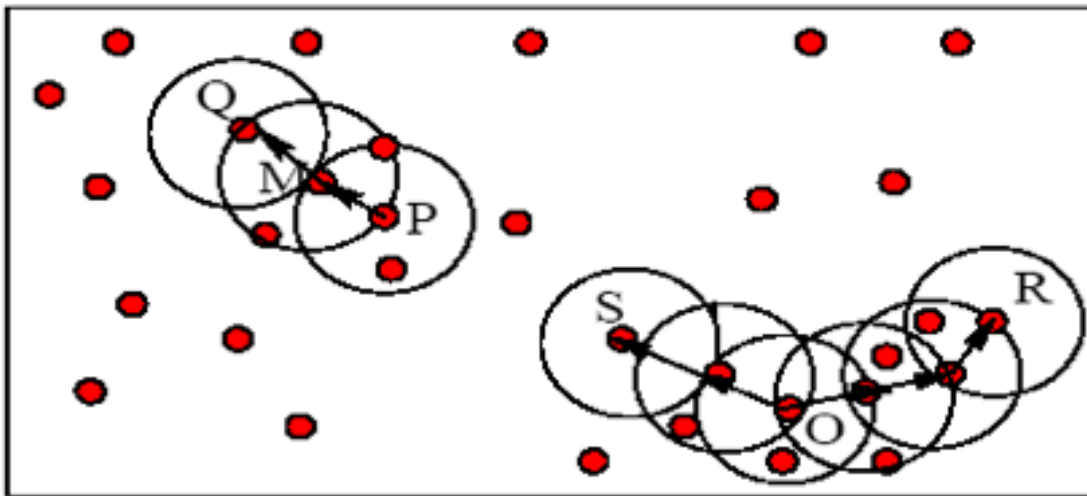


# DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: Randomly pick an unvisited point to form a new cluster and then spread to its density-reachable points



- MinPts = 3
- Solid rings show the  $\epsilon$  distance
- Un-clustered points: outliers

# DBSCAN: The Algorithm

**Algorithm: DBSCAN:** a density-based clustering algorithm.

**Input:**

- $D$ : a data set containing  $n$  objects,
- $\epsilon$ : the radius parameter, and
- $MinPts$ : the neighborhood density threshold.



# DBSCAN: The Algorithm

## Method:

- (1) mark all objects as **unvisited**;
- (2) **do**
- (3)     randomly select an unvisited object  $p$ ;
- (4)     mark  $p$  as **visited**;
- (5)     **if** the  $\epsilon$ -neighborhood of  $p$  has at least  $MinPts$  objects
- (6)         create a new cluster  $C$ , and add  $p$  to  $C$ ;
- (7)         let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $p$ ;
- (8)         **for** each point  $p'$  in  $N$
- (9)             **if**  $p'$  is **unvisited**
- (10)                 mark  $p'$  as **visited**;
- (11)                 **if** the  $\epsilon$ -neighborhood of  $p'$  has at least  $MinPts$  points,  
                    add those points to  $N$ ;
- (12)                 **if**  $p'$  is not yet a member of any cluster, add  $p'$  to  $C$ ;
- (13)         **end for**
- (14)         output  $C$ ;
- (15)     **else** mark  $p$  as **noise**;
- (16) **until** no object is **unvisited**;

# DBSCAN issue: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

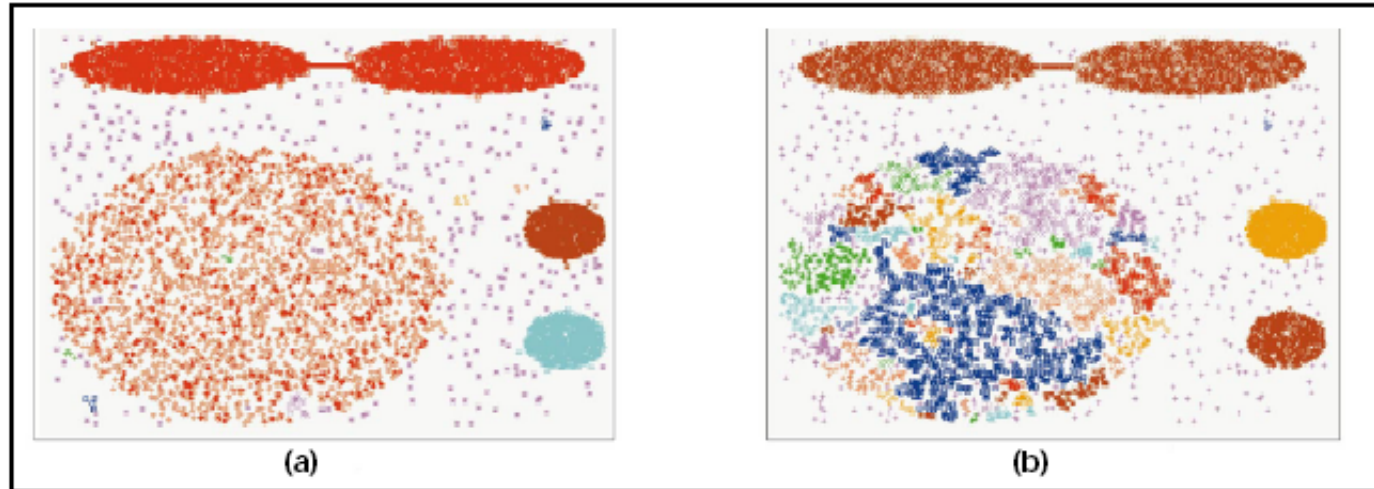
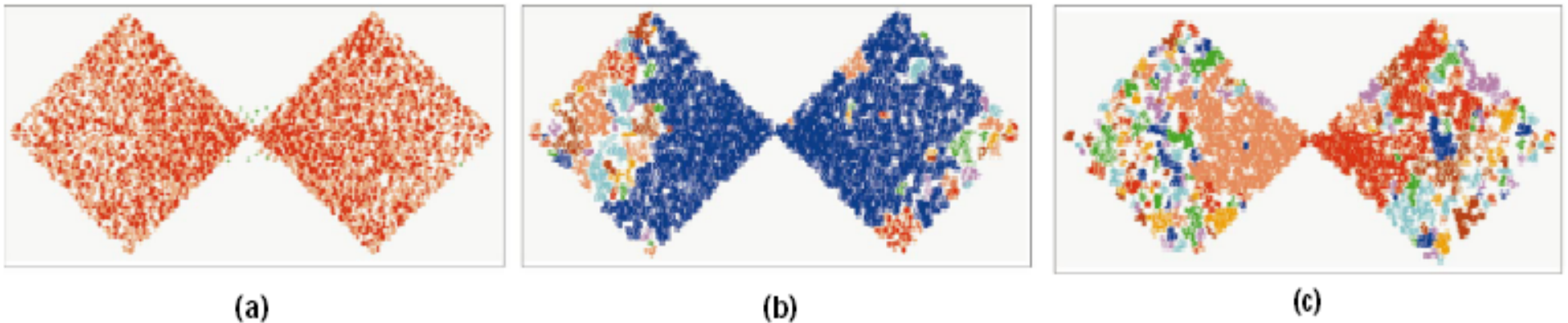


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

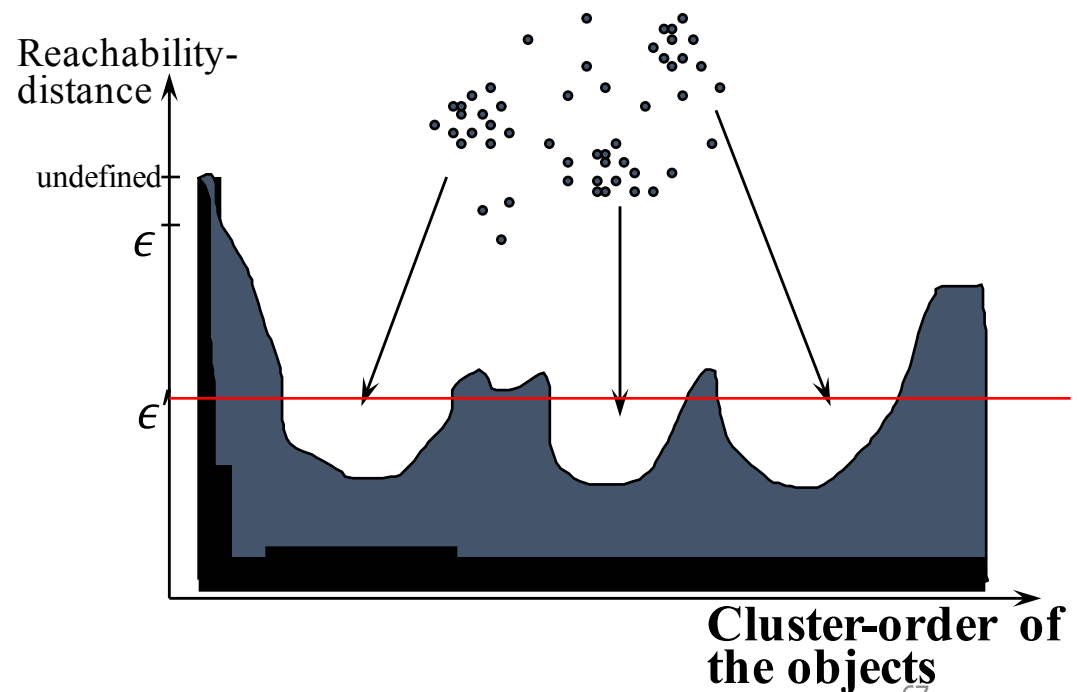


**DBSCAN online Demo:**

<http://webdocs.cs.ualberta.ca/~yaling/Cluster/Applet/Code/Cluster.html>

# OPTICS: A Cluster-Ordering Method (SIGMOD 1999) to alleviate parameter sensitivity issue of DBSCAN

- Parameters similar to DBSCAN:  $\epsilon$  and MinPts
- Extension from DBSCAN: Output clustering with a broad range of density levels (dynamic  $\epsilon$ ):
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically (figures)
- Observation: Given a MinPts, density-based clusters w.r.t. a higher density are completely contained in clusters w.r.t. to a lower density
- Idea: Higher-density points should be processed first—they are part of higher-density clusters
- OPTICS measures the density levels by a new concept, i.e., reachability distance.



# OPTICS order data points/objects by density level through reachability distance

- Core Distance of an object  $p$ : the smallest value  $\epsilon$  such that the  $\epsilon$ -neighborhood of  $p$  has at least  $\text{MinPts}$  objects
  - Let  $N_\epsilon(p)$ :  $\epsilon$ -neighborhood of  $p$ ,  $\epsilon$  is a distance value
  - Core-distance $_{\epsilon, \text{MinPts}}(p) =$ 
    - Undefined if  $\text{card}(N_\epsilon(p)) < \text{MinPts}$
    - $\text{MinPts}$ -distance( $p$ ), otherwise
- Reachability Distance of object  $p$  from core object  $q$  is the min radius value that makes  $p$  density-reachable from  $q$ 
  - Reachability-distance $_{\epsilon, \text{MinPts}}(p, q) =$ 
    - Undefined if  $q$  is not a core object
    - $\max(\text{core-distance}(q), \text{distance}(q, p))$ , otherwise

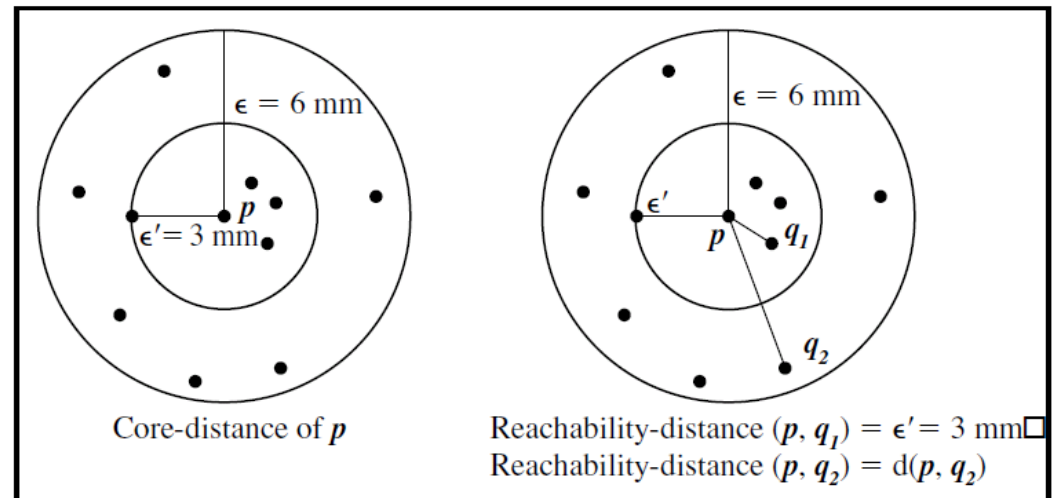
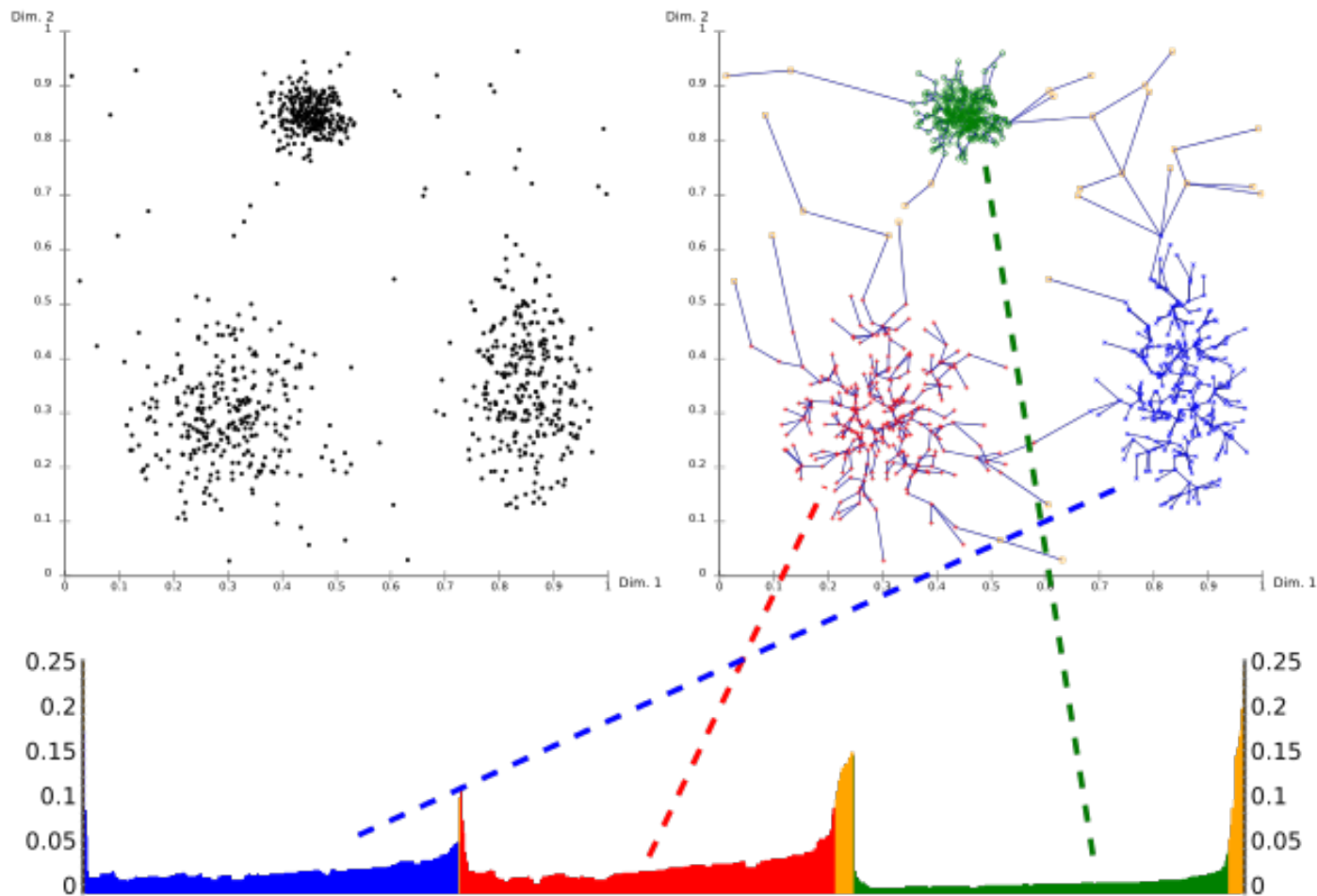
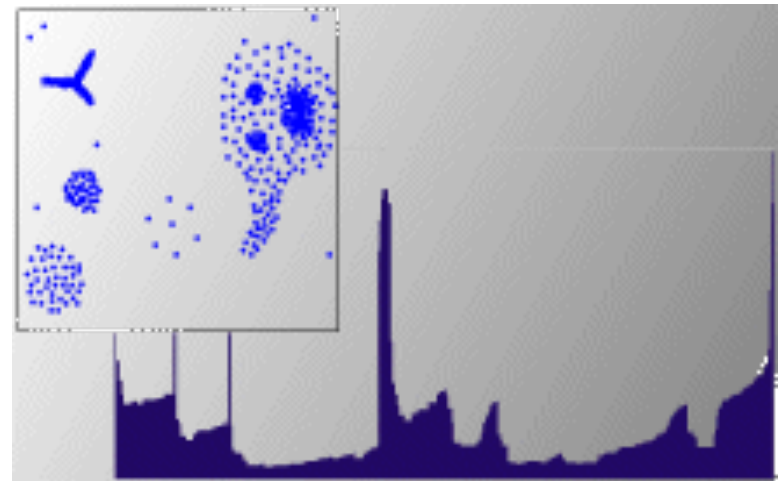
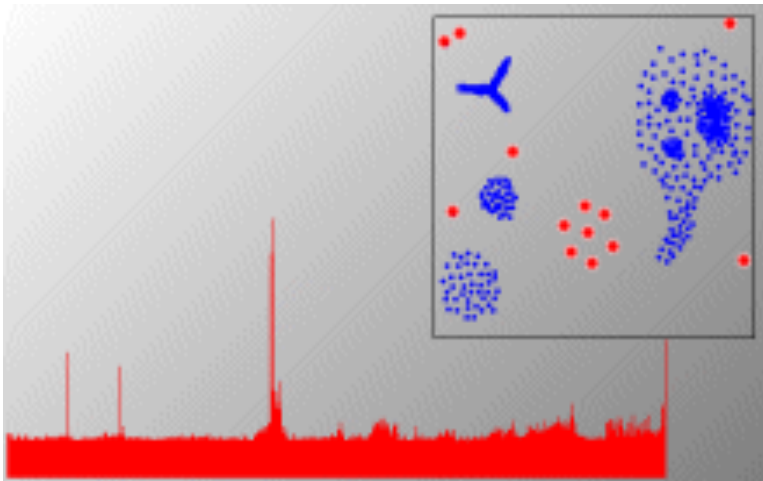


Figure 10.16: OPTICS terminology. Based on [ABKS99].


Output of OPTICS enables automatic and interactive cluster analysis (to infer the best eps)




OPTICS find nested structures with different parameter settings



# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering 
- Summary

# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering 
- Summary



# Determine the Number of Clusters

- Empirical method
  - # of clusters:  $k \approx \sqrt{\frac{n}{2}}$  for a dataset of  $n$  points, e.g.,  $n = 200$ ,  $k = 10$
- Elbow method
  - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- Cross validation method
  - Divide a given data set into  $m$  parts
  - Use  $m - 1$  parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any  $k > 0$ , repeat it  $m$  times, compare the overall quality measure w.r.t. different  $k$ 's, and find # of clusters that fits the data the best

# Measuring Clustering Quality

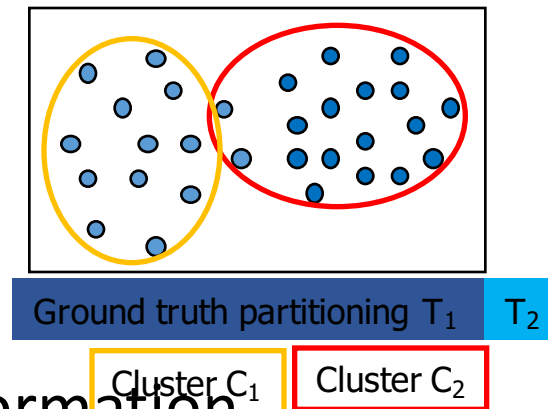
- 3 kinds of measures: External, internal and relative
- External: supervised, employ criteria not inherent to the dataset
  - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
- Internal: unsupervised, criteria derived from data itself
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are, e.g., Silhouette coefficient
- Relative: directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm

## Measuring Clustering Quality: External Methods

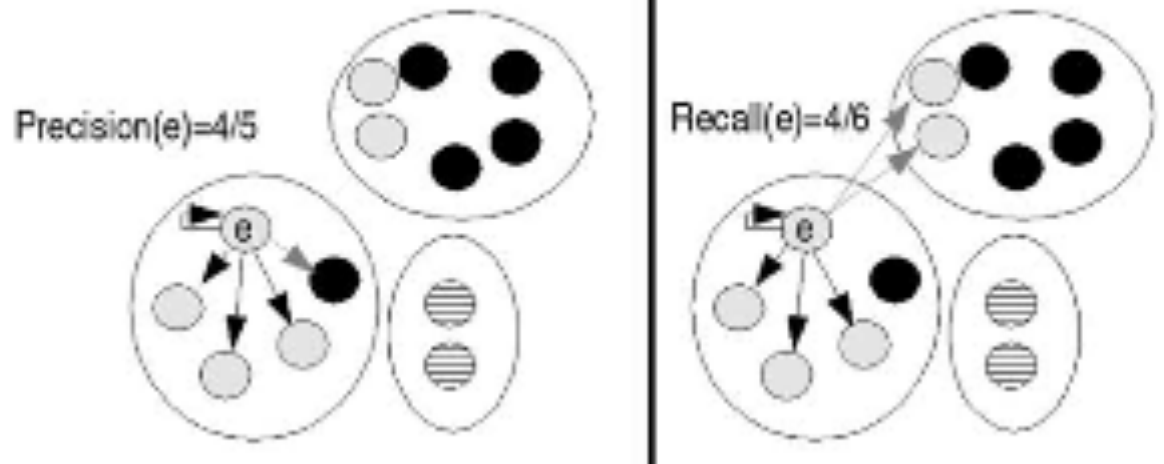
- Clustering quality measure:  $Q(C, T)$ , for a clustering  $C$  given the ground truth  $T$
- $Q$  is good if it satisfies the following **4** essential criteria
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., “miscellaneous” or “other” category)
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

## Some Commonly Used External Measures

- Matching-based measures
  - Purity, maximum matching, F-measure
- Entropy-Based Measures
  - Conditional entropy, normalized mutual information (NMI), variation of information
- Pair-wise measures
  - Four possibilities: True positive (TP), FN, FP, TN
  - Jaccard coefficient, Rand statistic, Fowlkes-Mallow measure
- Correlation measures
  - Discretized Huber static, normalized discretized Huber static



# B-Cubed Precision & Recall



$$Precision_i = \frac{\text{number of correct elements in the output chain containing entity}_i}{\text{number of elements in the output chain containing entity}_i}$$


$$Recall_i = \frac{\text{number of correct elements in the output chain containing entity}_i}{\text{number of elements in the truth chain containing entity}_i}$$

$$\text{Final Precision} = \sum_{i=1}^N w_i * Precision_i$$

$$\text{Final Recall} = \sum_{i=1}^N w_i * Recall_i$$

Note: The B-Cubed formula in the textbook is imprecise! Use this one (from [Bagga & Baldwin 1999])! 77

# Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary 

# Summary

- **Cluster analysis** groups objects based on their **similarity** and has wide applications
- Measure of similarity can be computed for **various types of data**
- Clustering algorithms can be **categorized** into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- **K-means** and **K-medoids** algorithms are popular partitioning-based clustering algorithms
- **Birch** and **Chameleon** are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- **DBSCAN**, **OPTICS**, and **DENCLU** are interesting density-based algorithms
- **STING** and **CLIQUE** are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways