

# CS145: INTRODUCTION TO DATA MINING

## 08: Vector Data: Clustering Basics

---

**Instructor: Yizhou Sun**

[yzsun@cs.ucla.edu](mailto:yzsun@cs.ucla.edu)


October 11, 2021

# Methods to Learn

|                         | Vector Data  | Set Data           | Sequence Data/Time Series | Text Data            | Graph Data          |
|-------------------------|--|--------------------|---------------------------|----------------------|---------------------|
| Classification          | Logistic Regression;<br>Decision Tree; NN            |                    |                           | Naïve Bayes for Text | Label Propagation   |
| Clustering              | K-means; Mixture Models                              |                    |                           | PLSA                 | Spectral Clustering |
| Prediction              | Linear Regression;<br>Regression Tree;<br>NN<br>GLM* |                    | AR Model                  |                      |                     |
| Frequent Pattern Mining |  | Apriori; FP growth | GSP; PrefixSpan           |                      |                     |
| Similarity Search       |  |                    | DTW                       |                      | P-PageRank          |
| Ranking                 |  |                    |                           |                      | PageRank            |

# Vector Data: Clustering Basics

---

- Clustering Analysis: Basic Concepts 
- Partitioning methods
- \*Hierarchical Methods
- \*Density-Based Methods
- Summary

# What is Cluster Analysis?

---

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation*, ...)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a **stand-alone tool** to get insight into data distribution
  - As a **preprocessing step** for other algorithms

# Applications of Cluster Analysis

---

- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those “far away” from any cluster


# Clustering: Application Examples

---

- **Biology:** taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- **Information retrieval:** document clustering
- **Land use:** Identification of areas of similar land use in an earth observation database
- **Marketing:** Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **City-planning:** Identifying groups of houses according to their house type, value, and geographical location
- **Earth-quake studies:** Observed earth quake epicenters should be clustered along continent faults
- **Climate:** understanding earth climate, find patterns of atmospheric and ocean

# Vector Data: Clustering Basics

---

- Clustering Analysis: Basic Concepts
- Partitioning methods 
- \*Hierarchical Methods
- \*Density-Based Methods
- Summary

# Partitioning Algorithms: Basic Concept

---

- Partitioning method: Partitioning a dataset  $D$  of  $n$  objects into a set of  $k$  clusters, such that the sum of squared distances is minimized (where  $c_j$  is the centroid or medoid of cluster  $C_j$ )

$$J = \sum_{j=1}^k \sum_{c(i)=j} d(x_i, c_j)^2$$

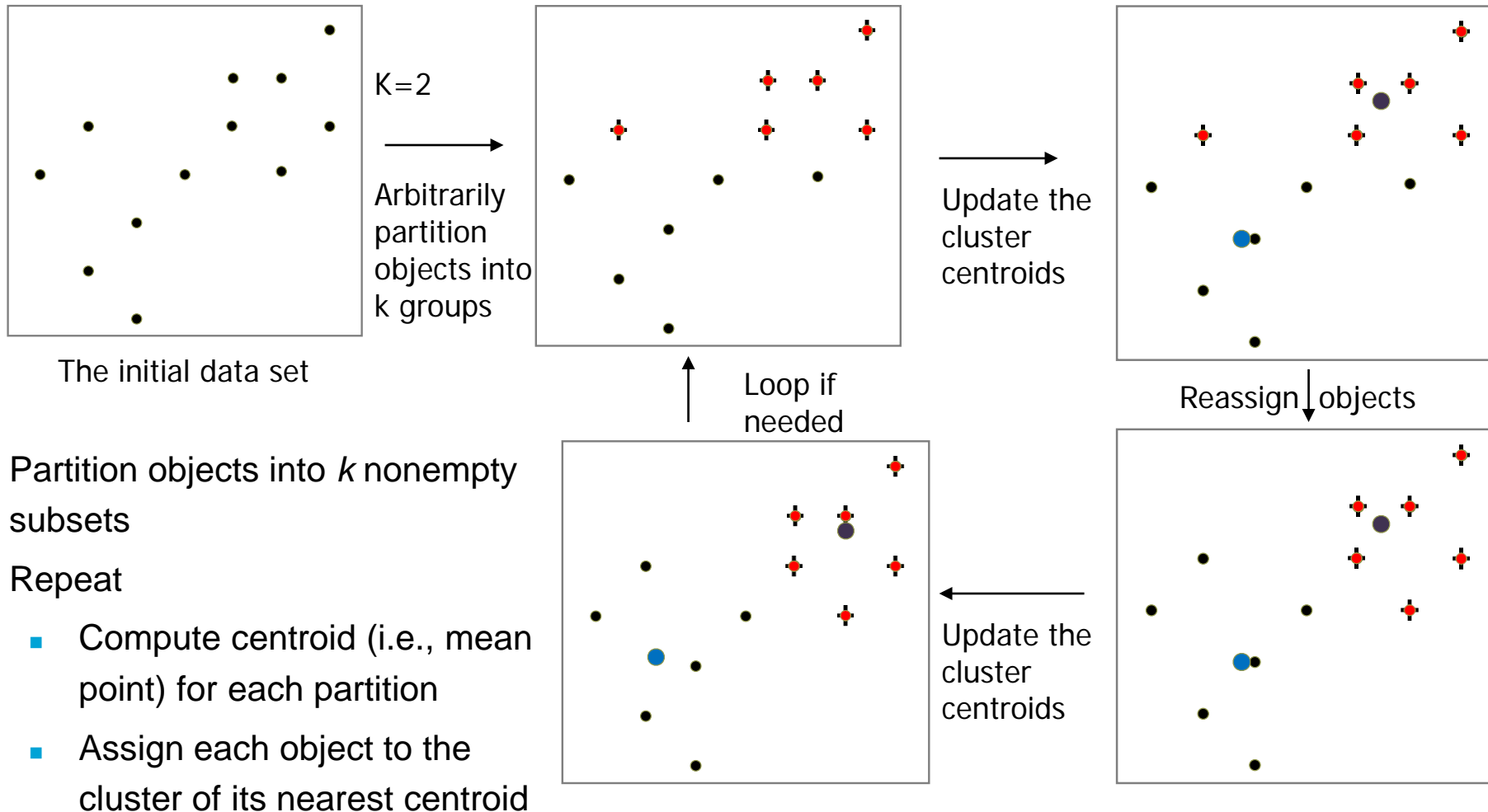
- Given  $k$ , find a partition of  $k$  clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

# The *K-Means* Clustering Method

---

- Given  $k$ , the *k-means* algorithm is implemented in four steps:
  - Step 0: Partition objects into  $k$  nonempty subsets
  - Step 1: Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Step 2: Assign each object to the cluster with the nearest seed point
  - Step 3: Go back to Step 1, stop when the assignment does not change

# An Example of *K-Means* Clustering



- Partition objects into  $k$  nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid

■ Until no change

# Theory Behind K-Means

---

- Objective function
  - $J = \sum_{j=1}^k \sum_{C(i)=j} \|x_i - c_j\|^2$
- Re-arrange the objective function
  - $J = \sum_{j=1}^k \sum_i w_{ij} \|x_i - c_j\|^2$ 
    - $w_{ij} \in \{0,1\}$
    - $w_{ij} = 1$ , if  $x_i$  belongs to cluster  $j$ ;  $w_{ij} = 0$ , otherwise
  - Looking for:
    - The best assignment  $w_{ij}$
    - The best center  $c_j$

# Solution of K-Means

---

- Iterations  $J = \sum_{j=1}^k \sum_i w_{ij} \|x_i - c_j\|^2$ 
  - Step 1: Fix centers  $c_j$ , find assignment  $w_{ij}$  that minimizes  $J$ 
    - $\Rightarrow w_{ij} = 1$ , if  $\|x_i - c_j\|^2$  is the smallest
  - Step 2: Fix assignment  $w_{ij}$ , find centers that minimize  $J$ 
    - $\Rightarrow$  first derivative of  $J = 0$
    - $\Rightarrow \frac{\partial J}{\partial c_j} = -2 \sum_i w_{ij} (x_i - c_j) = 0$
    - $\Rightarrow c_j = \frac{\sum_i w_{ij} x_i}{\sum_i w_{ij}}$ 
      - Note  $\sum_i w_{ij}$  is the total number of objects in cluster  $j$

slido



**Is K-means algorithm  
guaranteed to converge?**

ⓘ Start presenting to display the poll results on this slide.

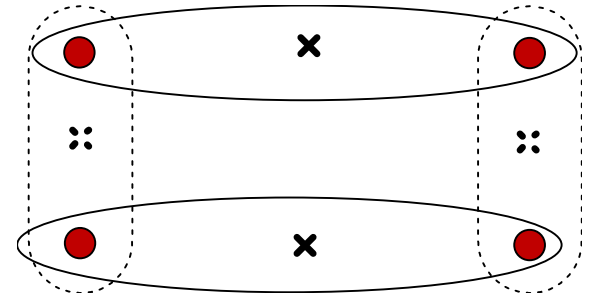
# Comments on the *K-Means* Method

---

- Strength: *Efficient*:  $O(tkn)$ , where  $n$  is # objects,  $k$  is # clusters, and  $t$  is # iterations. Normally,  $k, t \ll n$ .
- Comment: Often terminates at a *local optimal*
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify  $k$ , the *number* of clusters, in advance (there are ways to automatically determine the best  $k$  (see Hastie et al., 2009))
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

# Variations of the *K-Means* Method\*

- Most of the variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method




# The K-Medoid Clustering Method\*

---

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

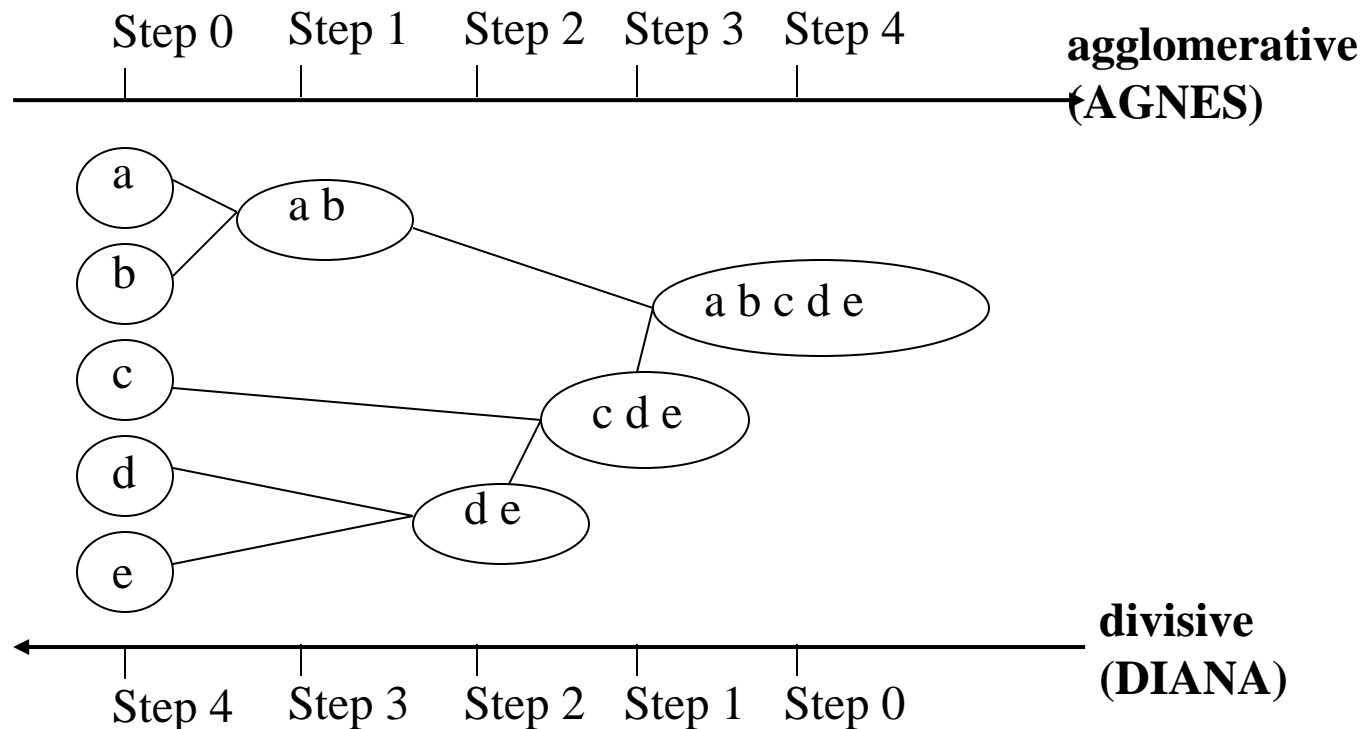
# Vector Data: Clustering Basics

---

- Clustering Analysis: Basic Concepts
- Partitioning methods
- \*Hierarchical Methods 
- \*Density-Based Methods
- Summary

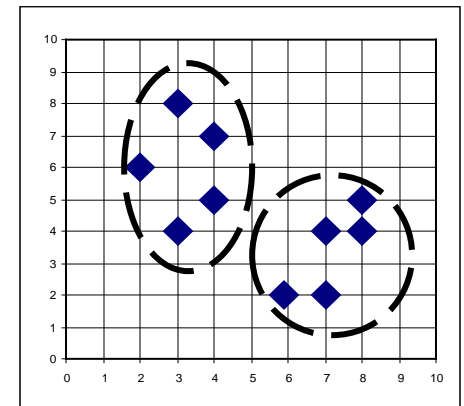
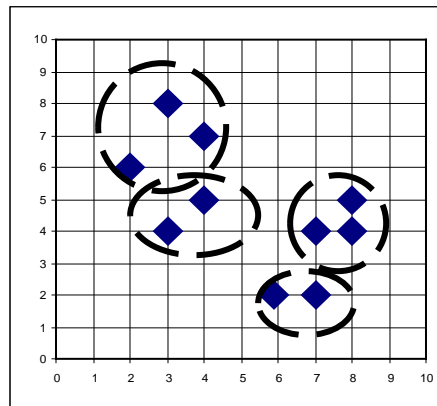
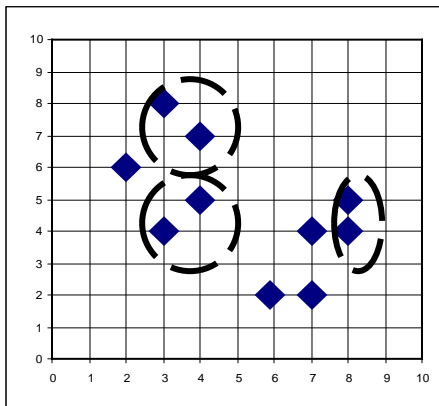
# Hierarchical Clustering

- 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



# Pseudo Code

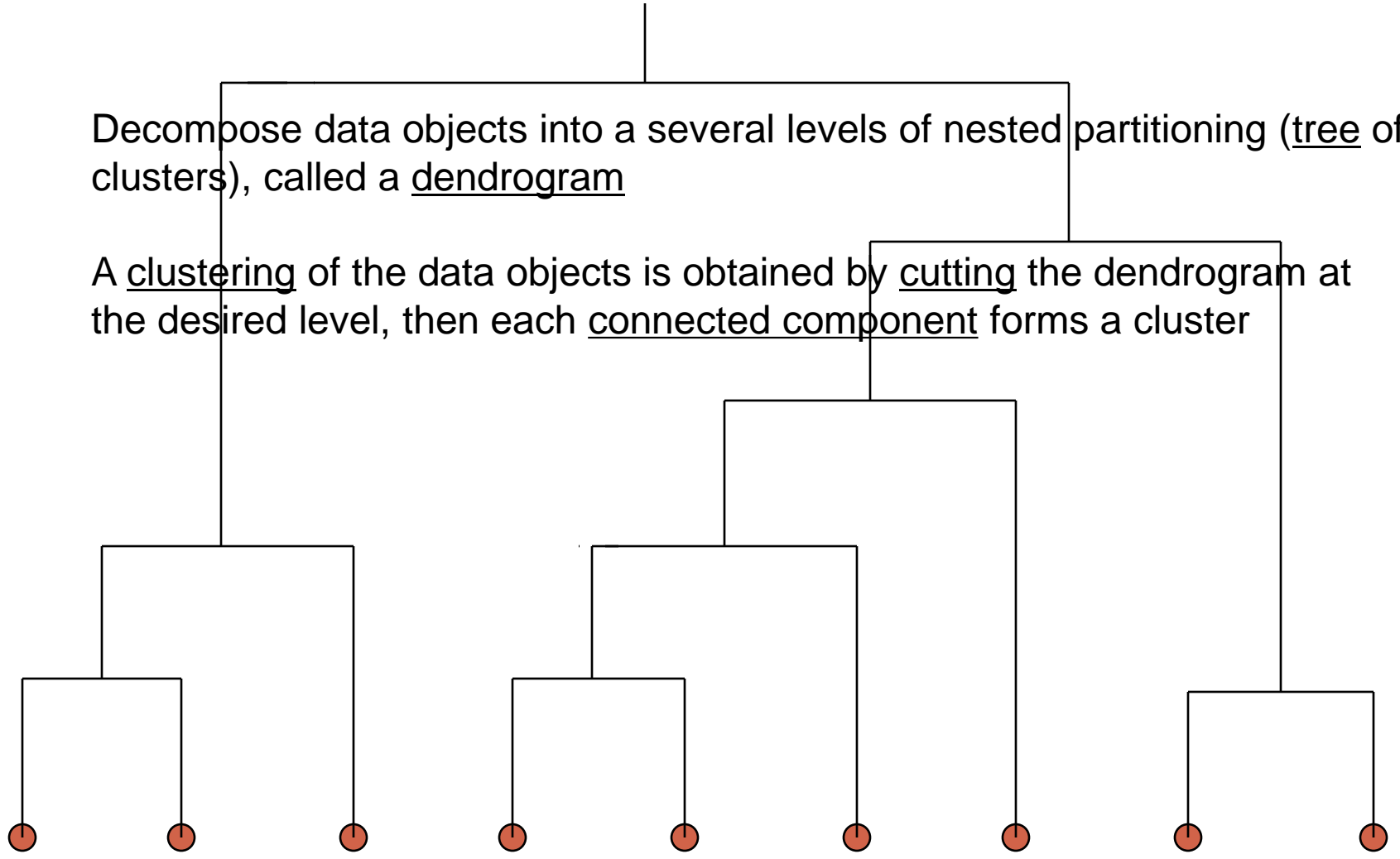
---

- Initialization: Place each data point into its own cluster and compute distance matrix between clusters
- Repeat:
  - Merge the two **closest** clusters
  - Update the distance matrix for the affected entries
- Until: all the data are merged into a single cluster

# Dendrogram: Shows How Clusters are Merged

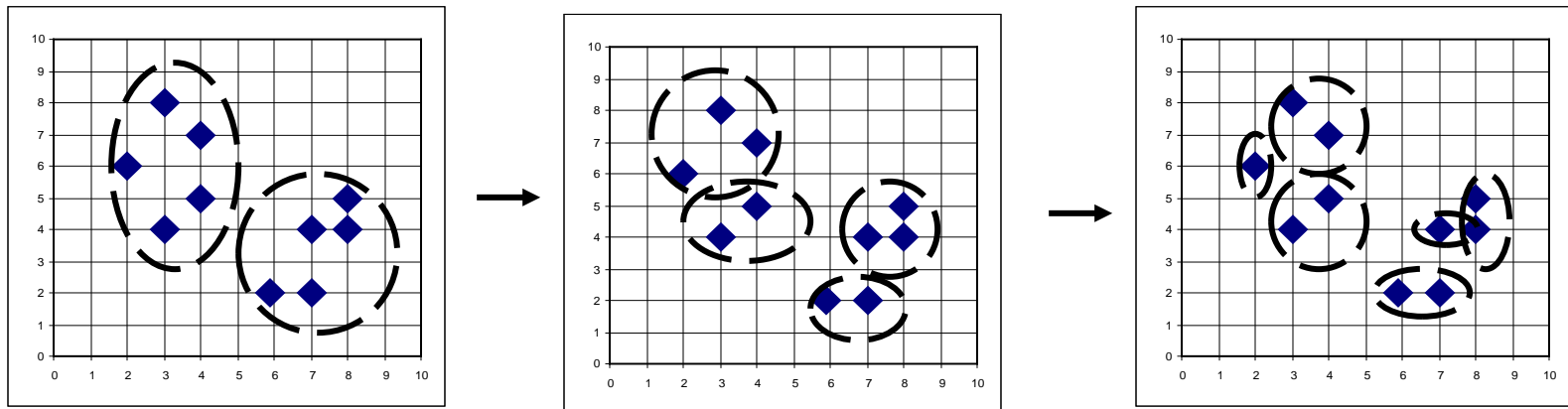
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

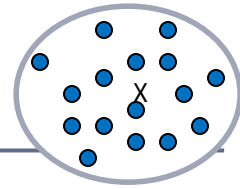
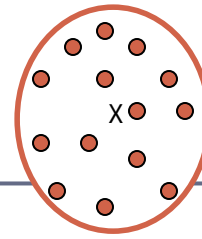


# DIANA (Divisive Analysis)

- 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

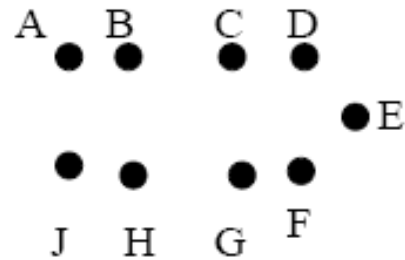


# Distance between Clusters

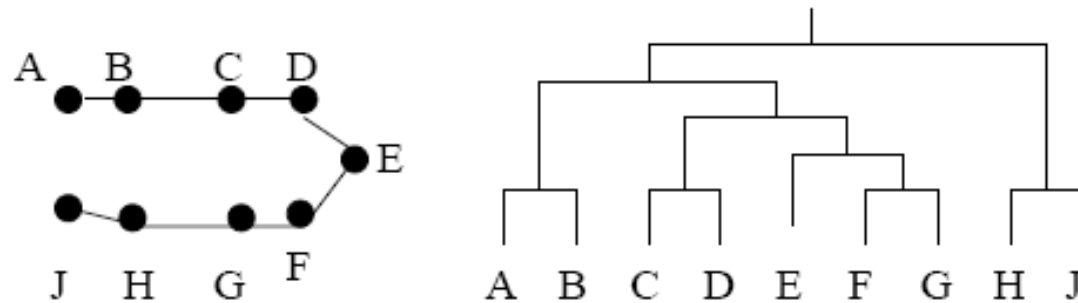


- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e.,  $\text{dist}(K_i, K_j) = \min \text{dist}(t_{ip}, t_{jq})$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e.,  $\text{dist}(K_i, K_j) = \max \text{dist}(t_{ip}, t_{jq})$
- **Average:** avg distance between an element in one cluster and an element in the other, i.e.,  $\text{dist}(K_i, K_j) = \text{avg dist}(t_{ip}, t_{jq})$
- **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

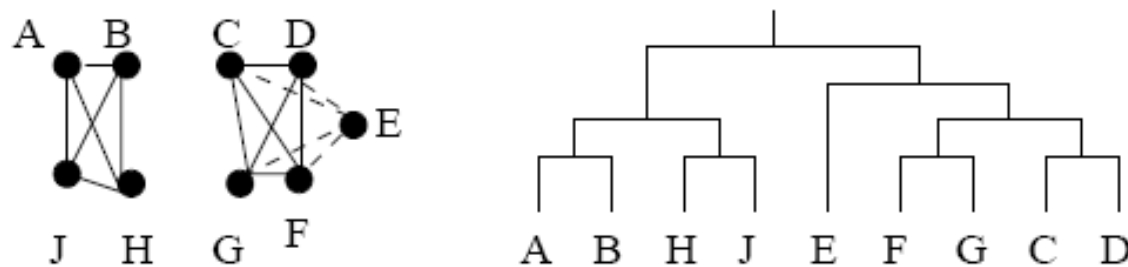
# Example: Single Link vs. Complete Link



(a) Data set



(b) Clustering using single linkage



(c) Clustering using complete linkage


# 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

# Vector Data: Clustering Basics

---

- Clustering Analysis: Basic Concepts
- Partitioning methods
- \*Hierarchical Methods
- \*Density-Based Methods 
- Summary

# Density-Based Clustering Methods

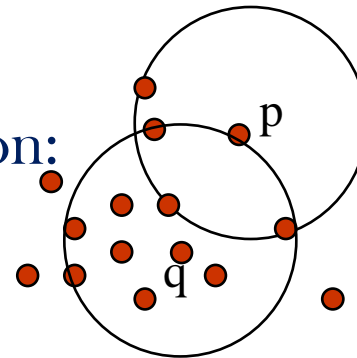
---

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- 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)

# DBSCAN: Basic Concepts

- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an *Eps*-neighborhood of that point
- $N_{Eps}(q)$ : {p belongs to D | dist(p,q) ≤ Eps}
- **Directly density-reachable**: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to  $N_{Eps}(q)$
  - *q* is a **core point**, **core point** condition:

$$|N_{Eps}(q)| \geq MinPts$$



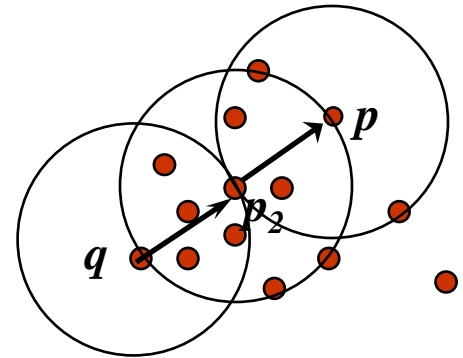
MinPts = 5

Eps = 1 cm

# Density-Reachable and Density-Connected

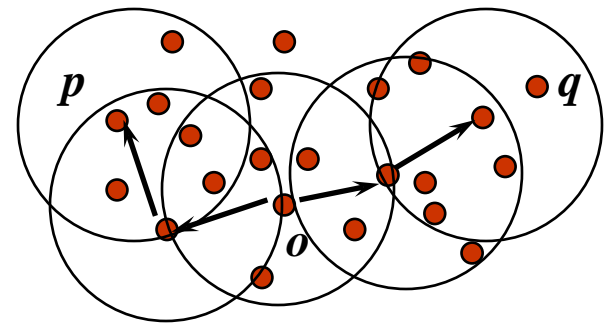
- Density-reachable:

- A point  $p$  is **density-reachable** from a point  $q$  w.r.t.  $Eps$ ,  $MinPts$  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  $p_i$



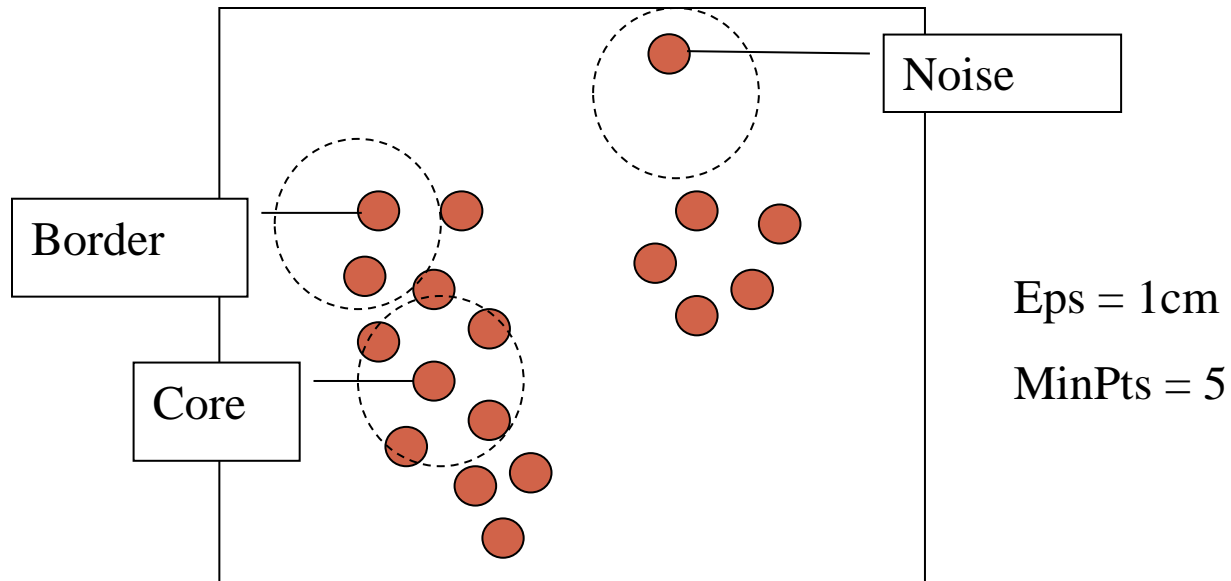
- Density-connected

- A point  $p$  is **density-connected** to a point  $q$  w.r.t.  $Eps$ ,  $MinPts$  if there is a point  $o$  such that both,  $p$  and  $q$  are density-reachable from  $o$  w.r.t.  $Eps$  and  $MinPts$



# 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
- **Noise**: object not contained in any cluster is noise
- Discovers clusters of arbitrary shape in spatial databases with noise



# DBSCAN: The Algorithm

---

- (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;

- *If a spatial index is used, the computational complexity of DBSCAN is  $O(n \log n)$ , where  $n$  is the number of database objects. Otherwise, the complexity is  $O(n^2)$*

# DBSCAN: Sensitive to Parameters

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

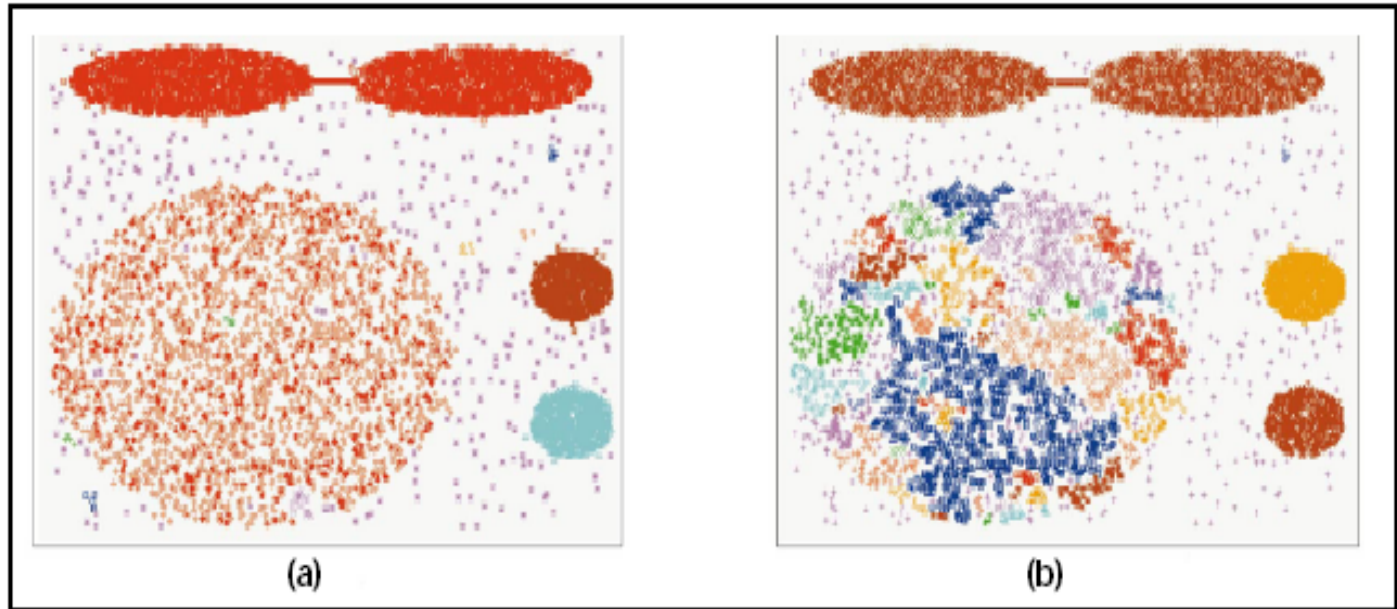
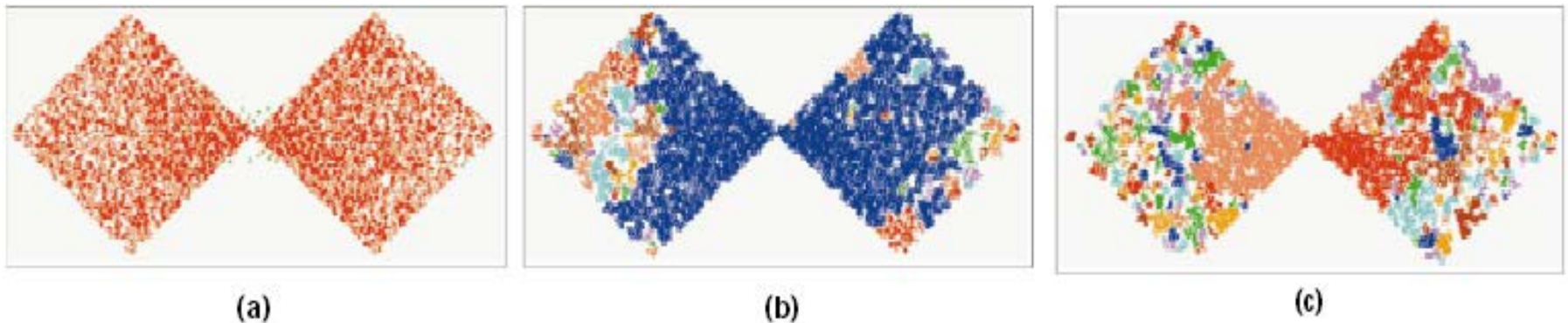


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN online Demo:

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


# Questions about Parameters

---

- Fix Eps, increase MinPts, what will happen?
- Fix MinPts, decrease Eps, what will happen?

# Vector Data: Clustering Basics

---

- Clustering Analysis: Basic Concepts
- Partitioning methods
- \*Hierarchical Methods
- \*Density-Based Methods
- 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**
- **K-means** and **K-medoids** algorithms are popular partitioning-based clustering algorithms
- **AGNES** and **DIANA** are interesting hierarchical clustering algorithms
- **DBSCAN**, **OPTICS\***, and **DENCLUE\*** are interesting density-based algorithms

# References (1)

---

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- Beil F., Ester M., Xu X.: "Frequent Term-Based Text Clustering", KDD'02
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB'98.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

## References (2)

---

- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *ICDE'99*, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.I A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.

## References (3)

---

- G. J. McLachlan and K.E. Bkaford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-Based Clustering in Large Databases, ICDT'01.
- A. K. H. Tung, J. Hou, and J. Han. Spatial Clustering in the Presence of Obstacles, ICDE'01
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets, SIGMOD' 02.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : An efficient data clustering method for very large databases. SIGMOD'96.
- Xiaoxin Yin, Jiawei Han, and Philip Yu, "[LinkClus: Efficient Clustering via Heterogeneous Semantic Links](#)", in Proc. 2006 Int. Conf. on Very Large Data Bases (VLDB'06), Seoul, Korea, Sept. 2006.

slido



## Audience Q&A Session

ⓘ Start presenting to display the audience questions on this slide.