COMP5331: Knowledge Discovery and Data Mining

Acknowledgement: Slides modified by Dr. Lei Chen based on the slides provided by Jiawei Han, Micheline Kamber, and Jian Pei

©2012 Han, Kamber & Pei. All rights reserved.

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

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

Clustering for Data Understanding and Applications

- 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
- Economic Science: market resarch

Clustering as a Preprocessing Tool (Utility)

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- 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

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
  - There is usually a separate “quality” function that measures the “goodness” of a cluster.
  - It is hard to define “similar enough” or “good enough”
    - The answer is typically highly subjective

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

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

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

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

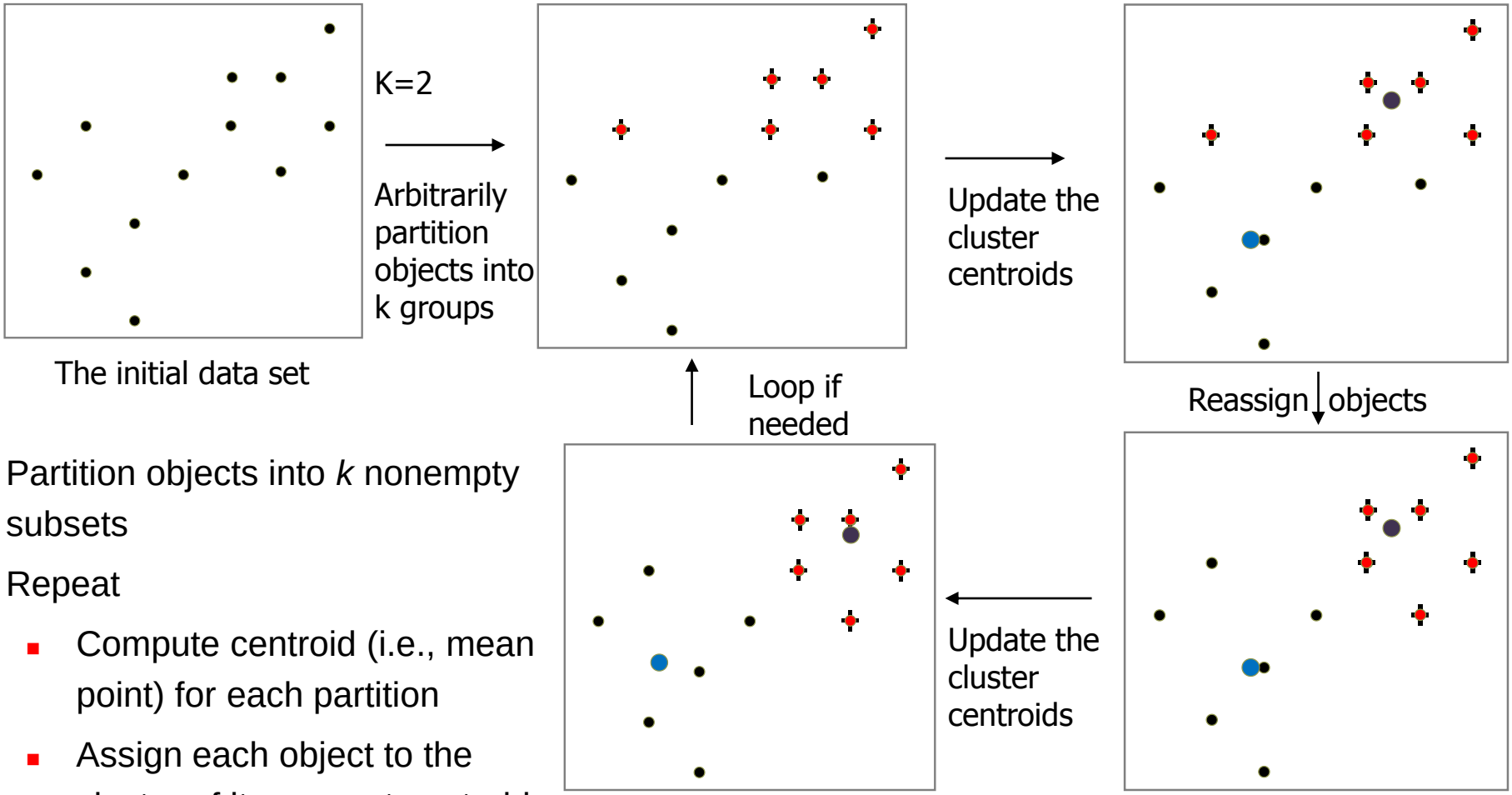
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (d(p, c_i))^2$$

- 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:
  - Partition objects into *k* nonempty subsets
  - 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)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

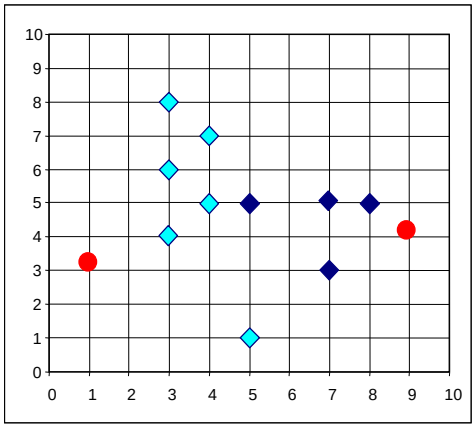
An Example of *K-Means* Clustering



- 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

The *K-Means* Clustering Method

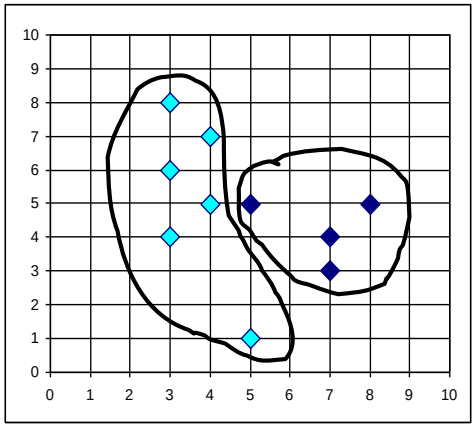
- Example



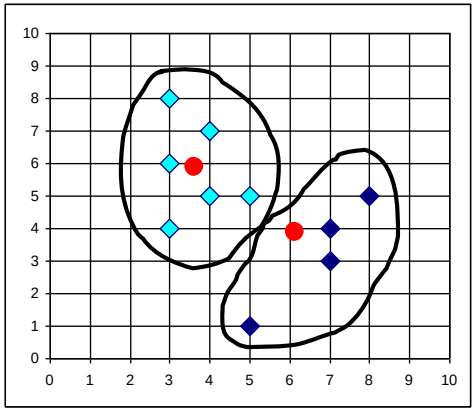
K=2

Arbitrarily choose K object as initial cluster center

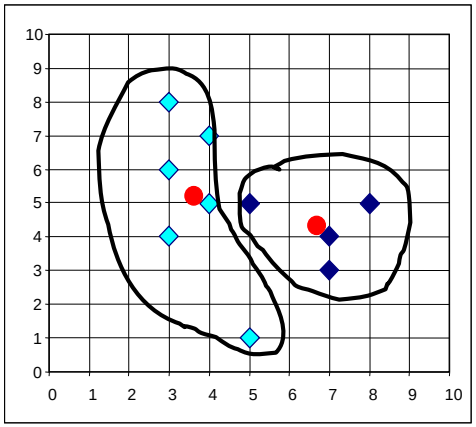
Assign each objects to most similar center



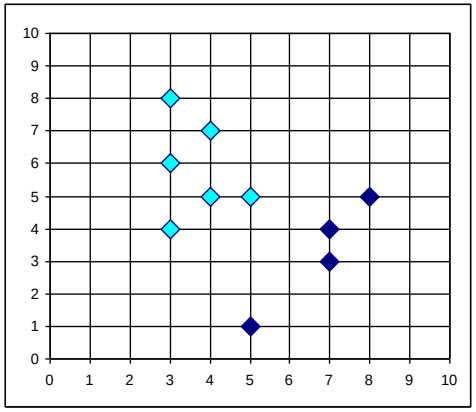
reassign



Update the cluster means



reassign



Update the cluster means

K-Means

- Consider the following 6 two-dimensional data points:
- x1: (0, 0), x2:(1, 0), x3(1, 1), x4(2, 1), x5(3, 1), x6(3, 0)
- If k=2, and the initial means are (0, 0) and (2, 1), (using Euclidean Distance)
- Use K-means to cluster the points.

K-Means

- Now we know the initial means:
- **Mean_one(0, 0)** and **mean_two(2, 1),**
- We are going to use Euclidean Distance to calculate the distance between each point and each mean.

- For example:
- **Point x1 (0, 0), point x1 is exactly the initial mean_one, so we can directly put x1 into cluster one.**

K-Means

- **Next we check Point x2 (1, 0)**

Distance1:

$$x2_and_mean_one = \sqrt[2]{(1-0)^2+(0-0)^2} = 1$$

Distance2:

$$x2_and_mean_two = \sqrt[2]{(1-2)^2+(0-1)^2} = \sqrt[2]{2}$$

Distance1<Distance2, so point x2 is closer to mean_one,

Thus x2 belongs to cluster one.

K-Means

Similarly for point x3 (1, 1),

Distance1:

$$x3_and_mean_one = \sqrt[2]{(1-0)^2+(1-0)^2} = \sqrt[2]{2}$$

Distance2:

$$x3_and_mean_two = \sqrt[2]{(1-2)^2+(1-1)^2} = 1$$

Distance1>Distance2, so point x3 is closer to mean_two,

Thus x3 belongs to cluster two.

K-Means

Next point x4 (2, 1),

- **point x4 is exactly the initial mean_two, so we can directly put x4 into cluster two.**

Next point x5 (3, 1),

Distance1:

$$x5_and_mean_one = \sqrt[2]{(3-0)^2+(1-0)^2} = \sqrt[2]{10}$$

Distance2:

$$x5_and_mean_two = \sqrt[2]{(3-2)^2+(1-1)^2} = 1$$

Distance1>Distance2, so point x5 is closer to mean_two,

Thus x5 belongs to cluster two.

K-Means

Next point x6 (3, 0),

Distance1:

x6_and_mean_one = $\sqrt[2]{(3-0)^2+(0-0)^2} = \sqrt[2]{9}$

Distance2:

x6_and_mean_two = $\sqrt[2]{(3-2)^2+(0-1)^2} = \sqrt[2]{2}$

Distance1>Distance2, so point x6 is closer to mean_two,

Thus x6 belongs to cluster two.

K-Means

Now the two clusters are as follows:

Cluster_One:

x1(0, 0), x2(1, 0)

Cluster_Two:

x3(1, 1), x4(2, 1), x5(3, 1), x6(3, 0)

Now update the cluster means:

Mean of Cluster_One:

mean_one = ((0+1)/2, (0+0)/2) = (0.5, 0)

Mean of Cluster_Two:

mean_two = ((1+2+3+3)/4, (1+1+1+0)/4) = (2.25, 0.75)

K-Means

Now **re-assign the points** according to the new means:

mean_one(0.5, 0) and mean_two (2.25, 0.75)

For example, x1 (0,0)

Distance1:

$$x1_and_mean_one = \sqrt[2]{(0-0.5)^2+(0-0)^2} = \sqrt[2]{0.25}$$

Distance2:

$$x1_and_mean_two = \sqrt[2]{(0-2.25)^2+(0-0.75)^2} = \sqrt[2]{5.625}$$

Distance1 < Distance2, So point x1 should remain in the cluster one.

Repeat the same for the other data points.

K-Means

After this **re-assign iteration:**

Cluster One:

x1(0, 0), x2(1, 0), x3(1, 1)

Cluster Two:

x4(2, 0), x5(3, 1), x6(3, 0)

Now update the cluster means:

Mean of Cluster_One:

mean_one = ((0+1+1)/3, (0+0+1)/3) = (0.667, 0.33)

Mean of Cluster_Two:

mean_two = ((2+3+3)/3, (0+1+0)/3) = (2.67, 0.33)

K-Means

Now **re-assign the points** according to the **new** means:

mean_one(0.667, 0.33) and mean_two (2.67, 0.33)

…

However, after the re-assign process, we find that the clusters remain the same, **so we can stop here since there is no change at all.**

So the final results are :

Cluster One:

x1(0, 0), x2(1, 0), x3(1, 1)

Cluster Two:

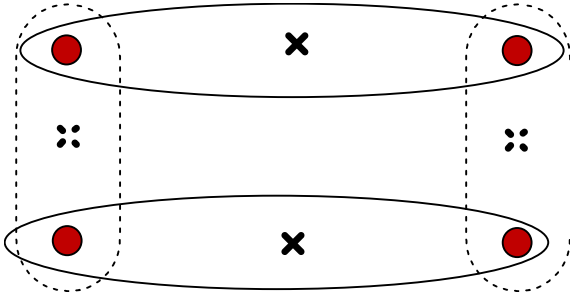
x4(2, 0), x5(3, 1), x6(3, 0)

Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- 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

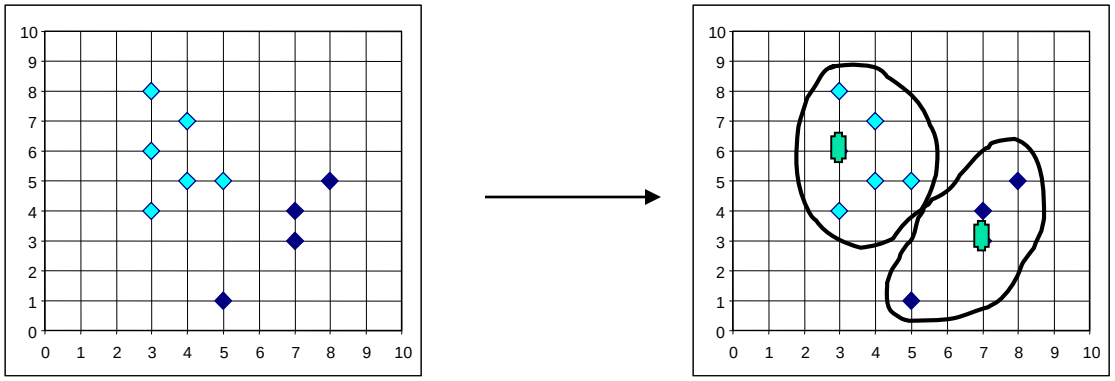
- Most of the variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means



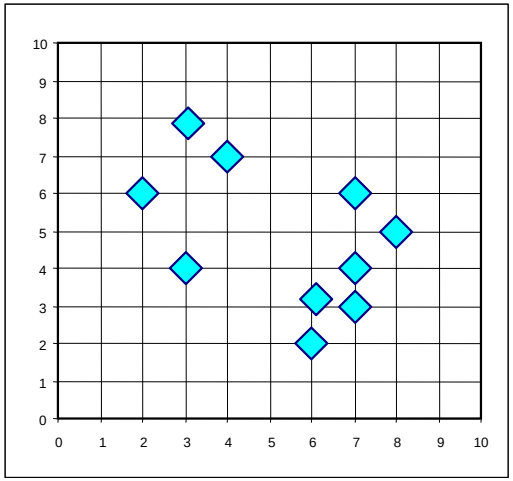
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

What Is the Problem of the K-Means Method?

- 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

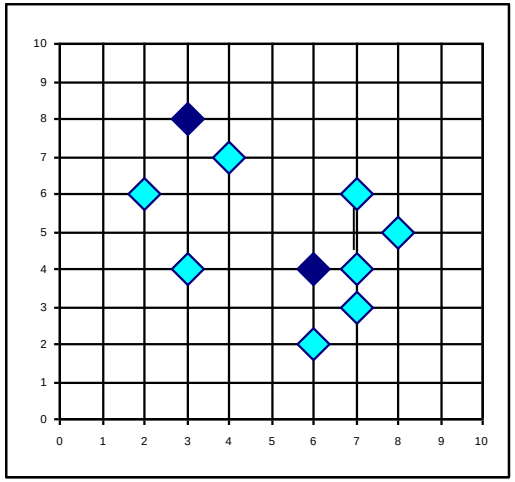


PAM: A Typical K-Medoids Algorithm



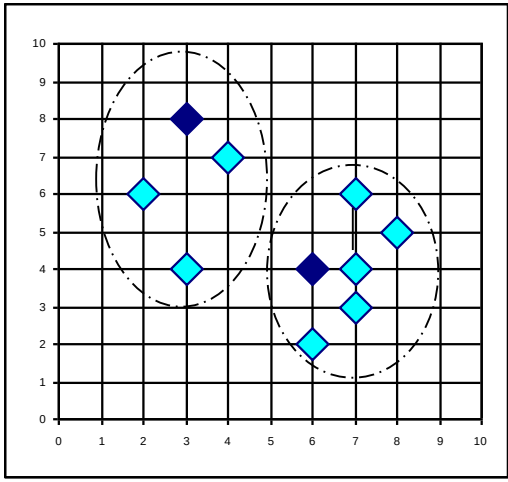
K=2

Arbitrary choose k object as initial medoids

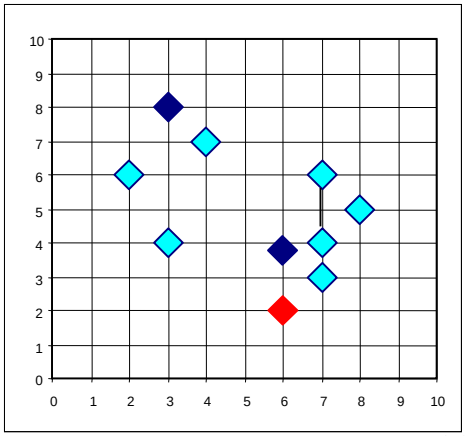


Assign each remaining object to nearest medoids

Total Cost = 20

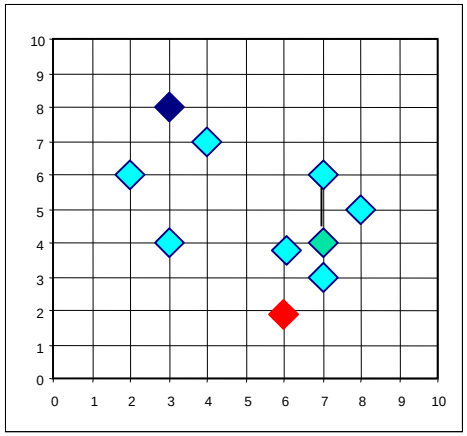


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

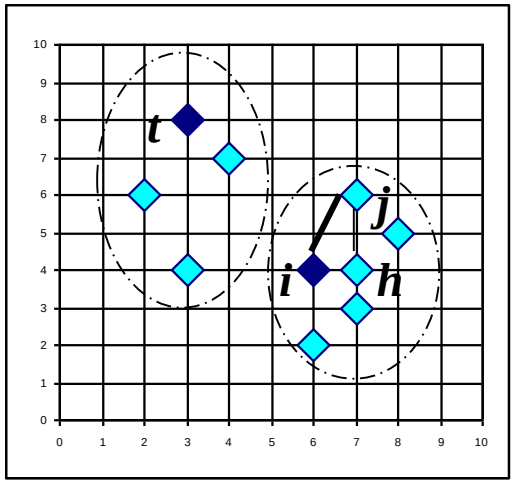
Do loop

Until no change

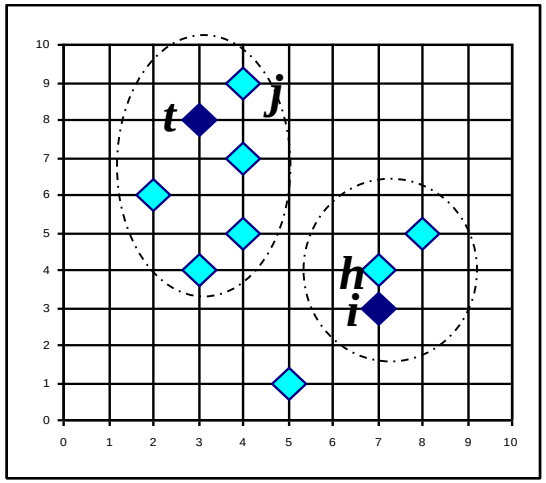
PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih} < 0$, ***i*** is replaced by ***h***
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change

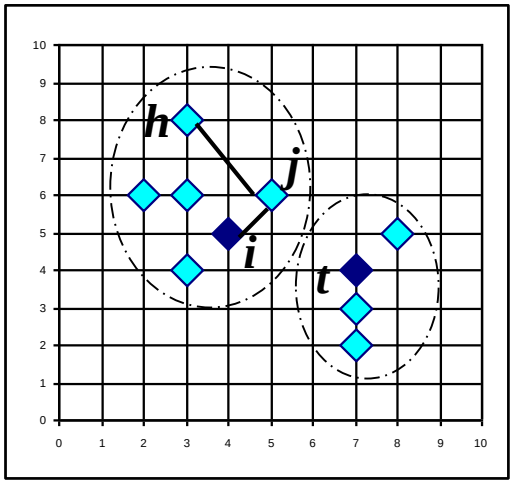
PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$



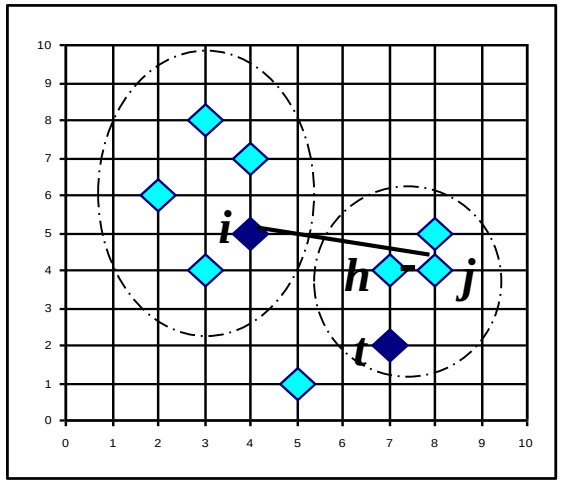
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



$C_{jih} = d(j, t) - d(j, i)$

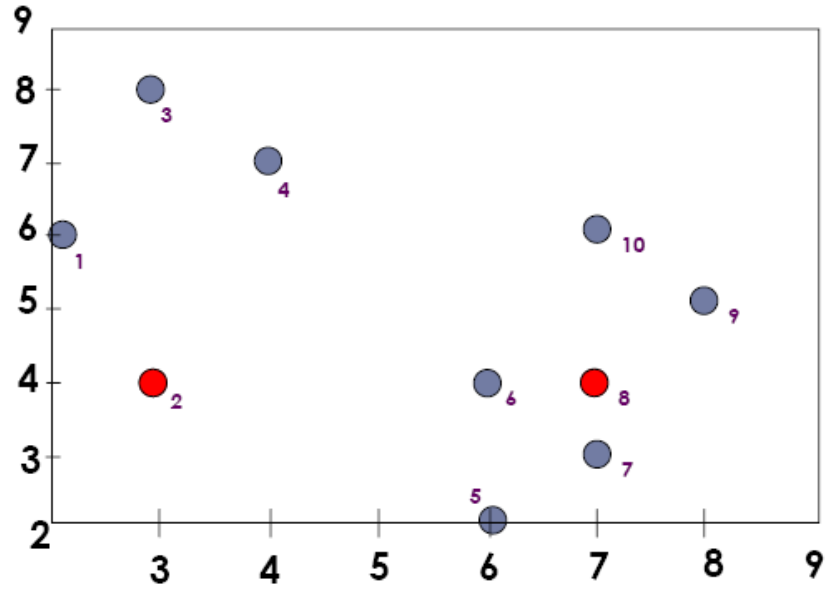


$C_{jih} = d(j, h) - d(j, t)$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



Goal: create two clusters

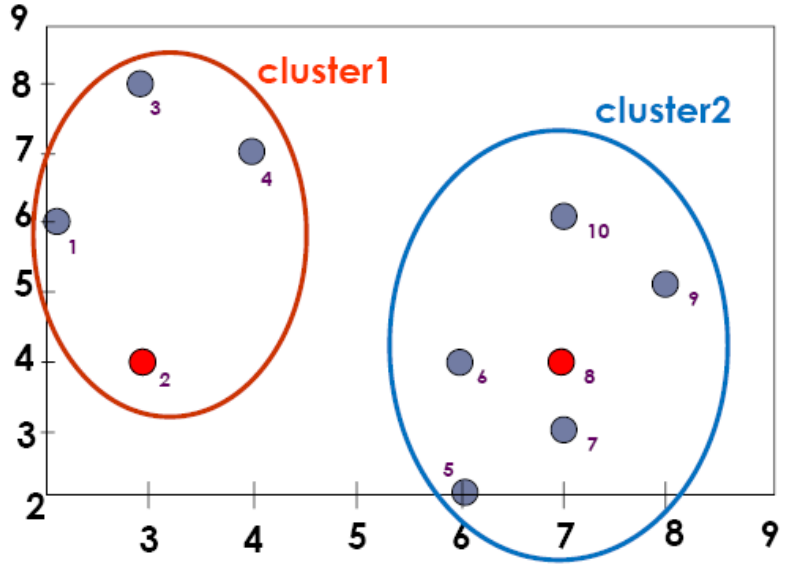
Choose randmly two medoids

$O_2 = (3,4)$
$O_8 = (7,4)$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Assign each object to the closest representative object

→Using L1 Metric (Manhattan), we form the following clusters

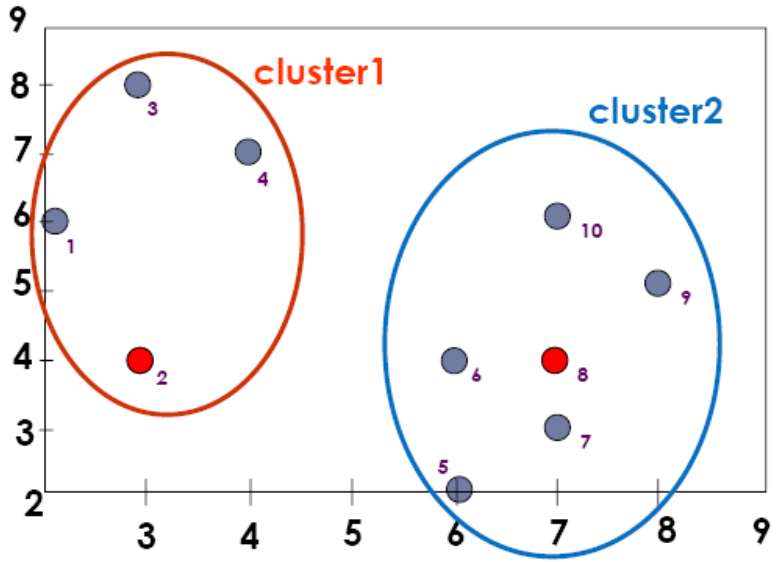
Cluster1 = {O_1, O_2, O_3, O_4}

Cluster2 = {O_5, O_6, O_7, O_8, O_9, O_{10}}

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the absolute error criterion **[for the set of Medoids (O2,O8)]**

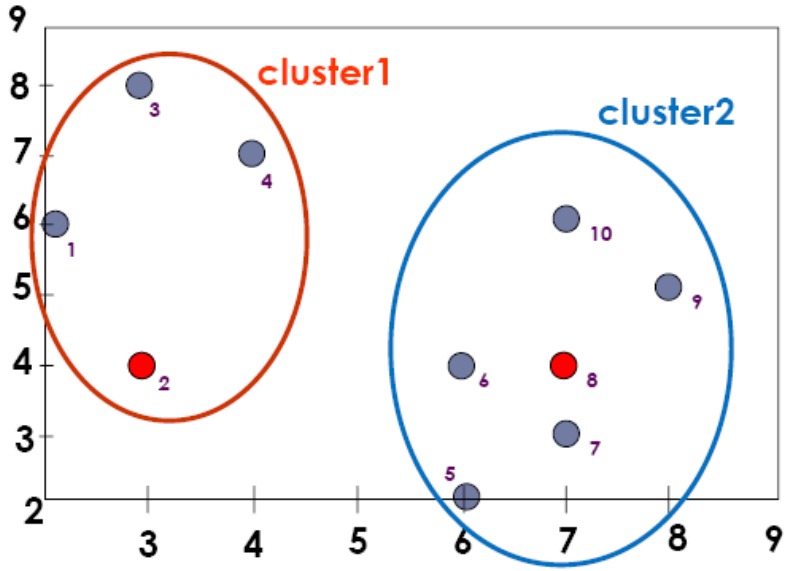
$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



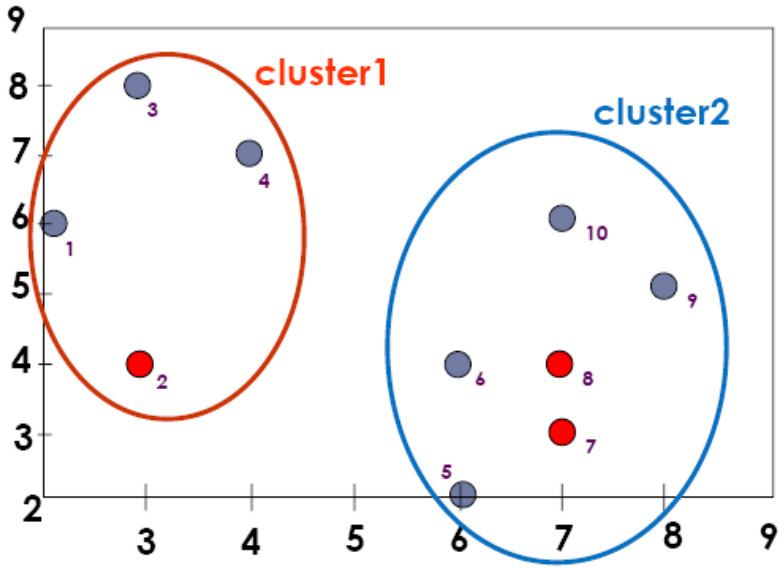
→The absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E = (3+4+4)+(3+1+1+2+2) = 20$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Choose a random object O_7

→Swap O8 and O7

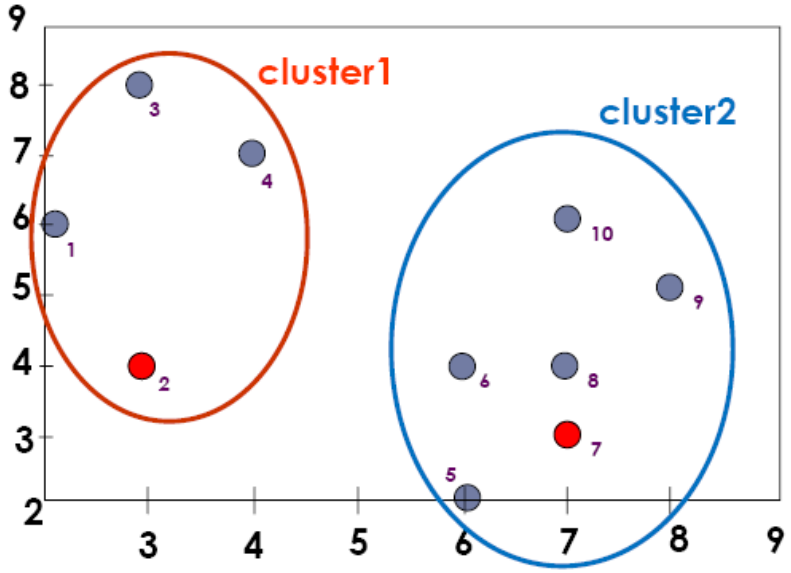
→Compute the absolute error criterion **[for the set of Medoids (O2,O7)]**

$$E = (3+4+4)+(2+2+1+3+3) = 22$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the cost function

Absolute error [for O_2,O_7] – Absolute error [O_2,O_8]

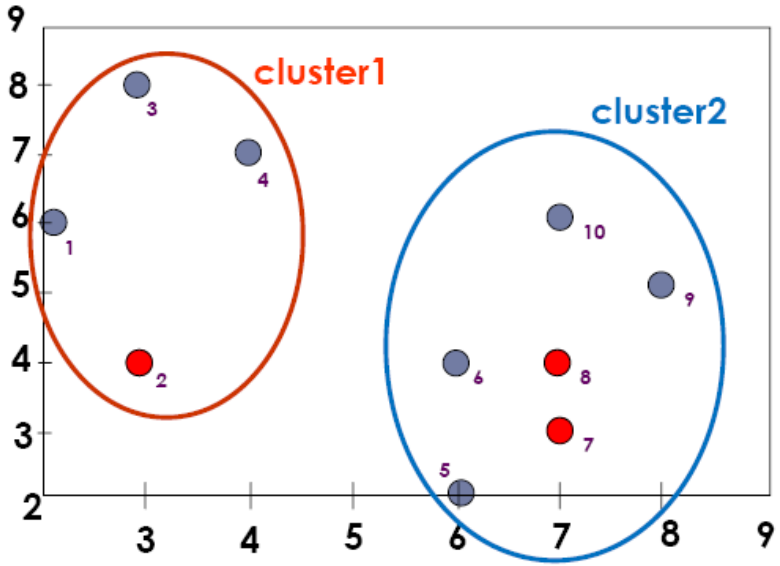
$$S = 22 - 20$$

S> 0 ⇒ it is a bad idea to replace O_8 by O_7

K-Medoids Method

Data Objects

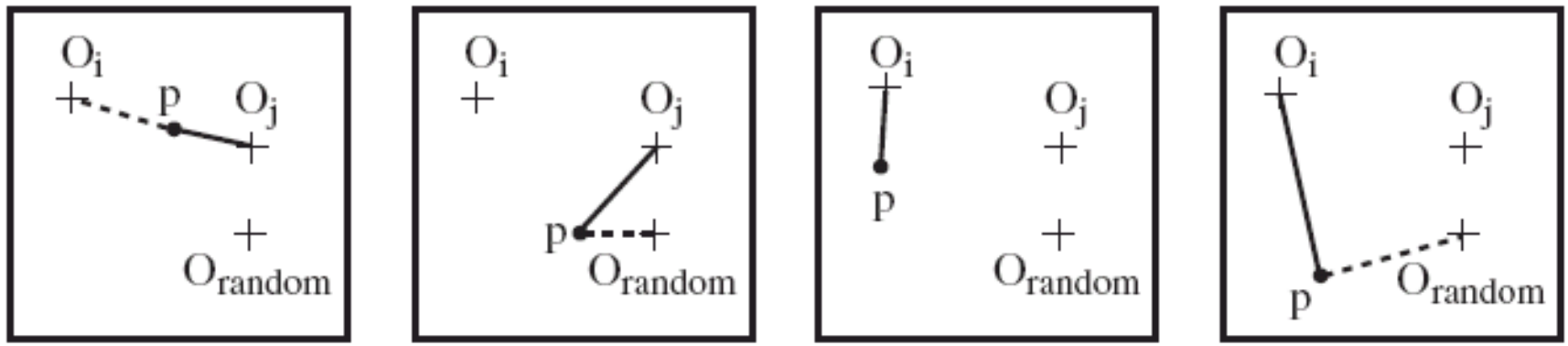
	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



- In this example, changing the medoid of cluster 2 did not change the assignments of objects to clusters.
- What are the possible cases when we replace a medoid by another object?

PAM Clustering: Finding the Best Cluster Center

- Case 1: p currently belongs to o_j. If o_j is replaced by o_{random} as a representative object and p is the closest to one of the other representative object o_i, then p is reassigned to o_i



1. Reassigned to O_i
2. Reassigned to O_{random}
3. No change
4. Reassigned to O_{random}

- data object
+ cluster center
— before swapping
--- after swapping

What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where n is # of data,k is # of clusters

➔ Sampling-based method

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as SPlus
  - It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS ("Randomized" CLARA) *(1994)*

- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
  - Draws sample of neighbors dynamically
  - The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
  - If the local optimum is found, *it* starts with new randomly selected node in search for a new local optimum
- Advantages: More efficient and scalable than both *PAM* and *CLARA*
- Further improvement: Focusing techniques and spatial access structures (Ester et al.'95)

The K-Medoid Clustering Method

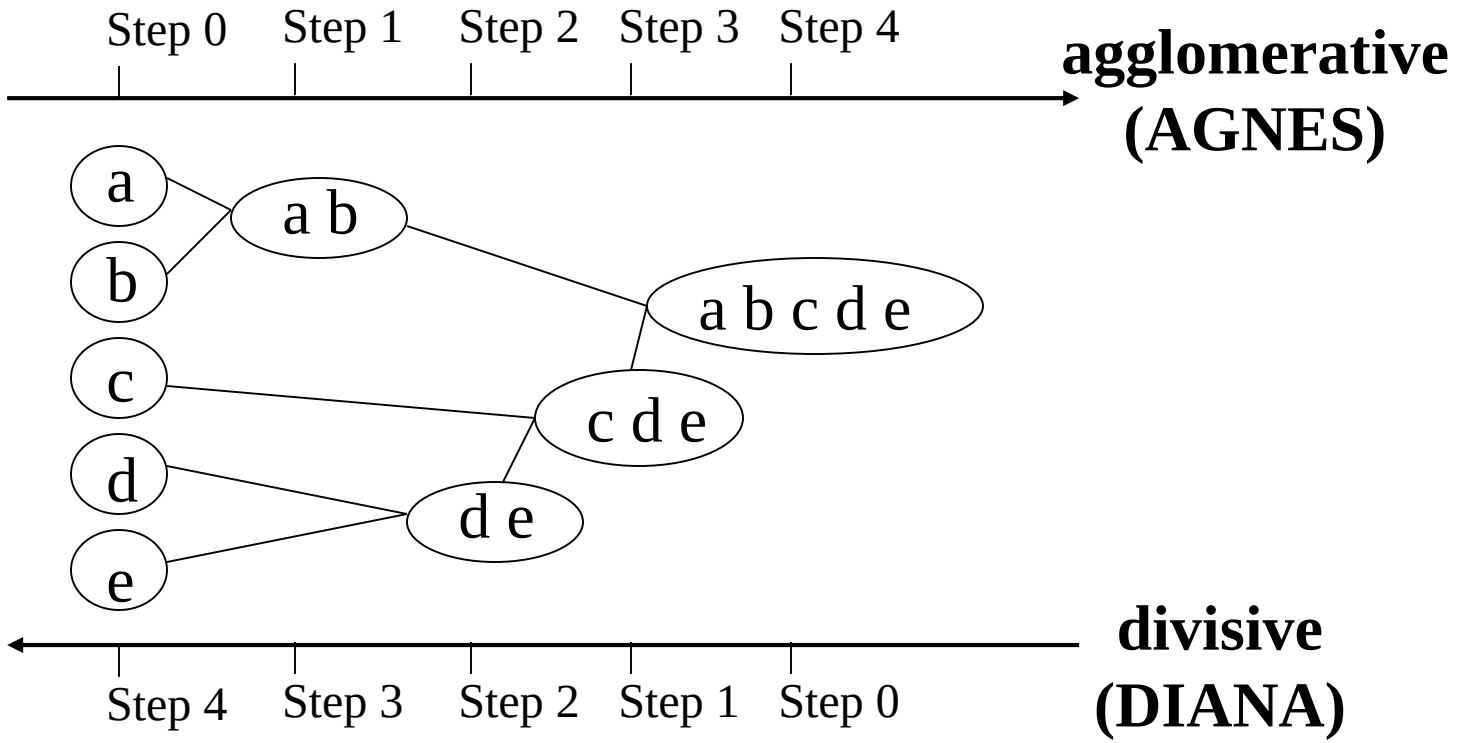
- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting 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

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

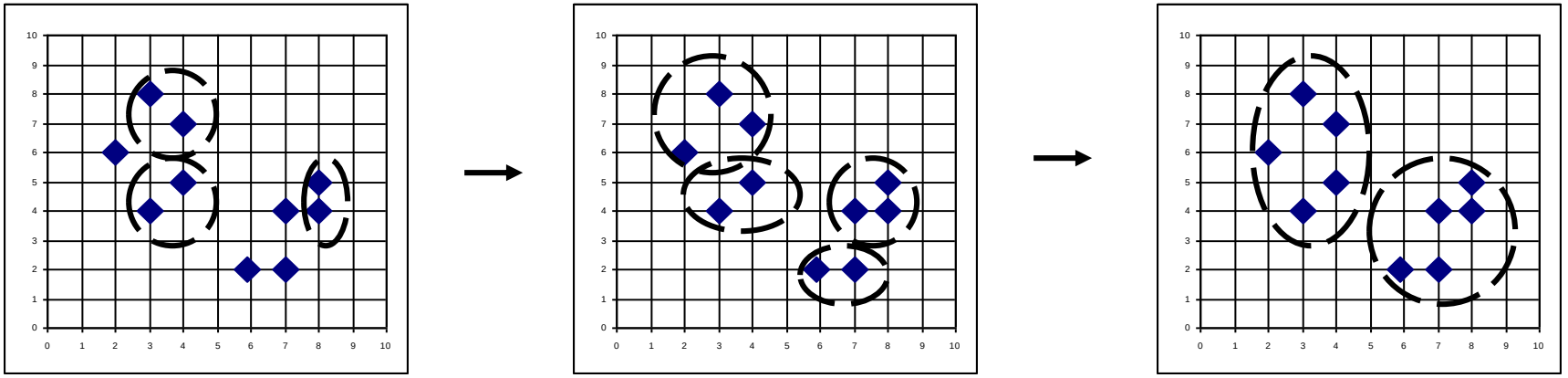
Hierarchical Clustering

- 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)

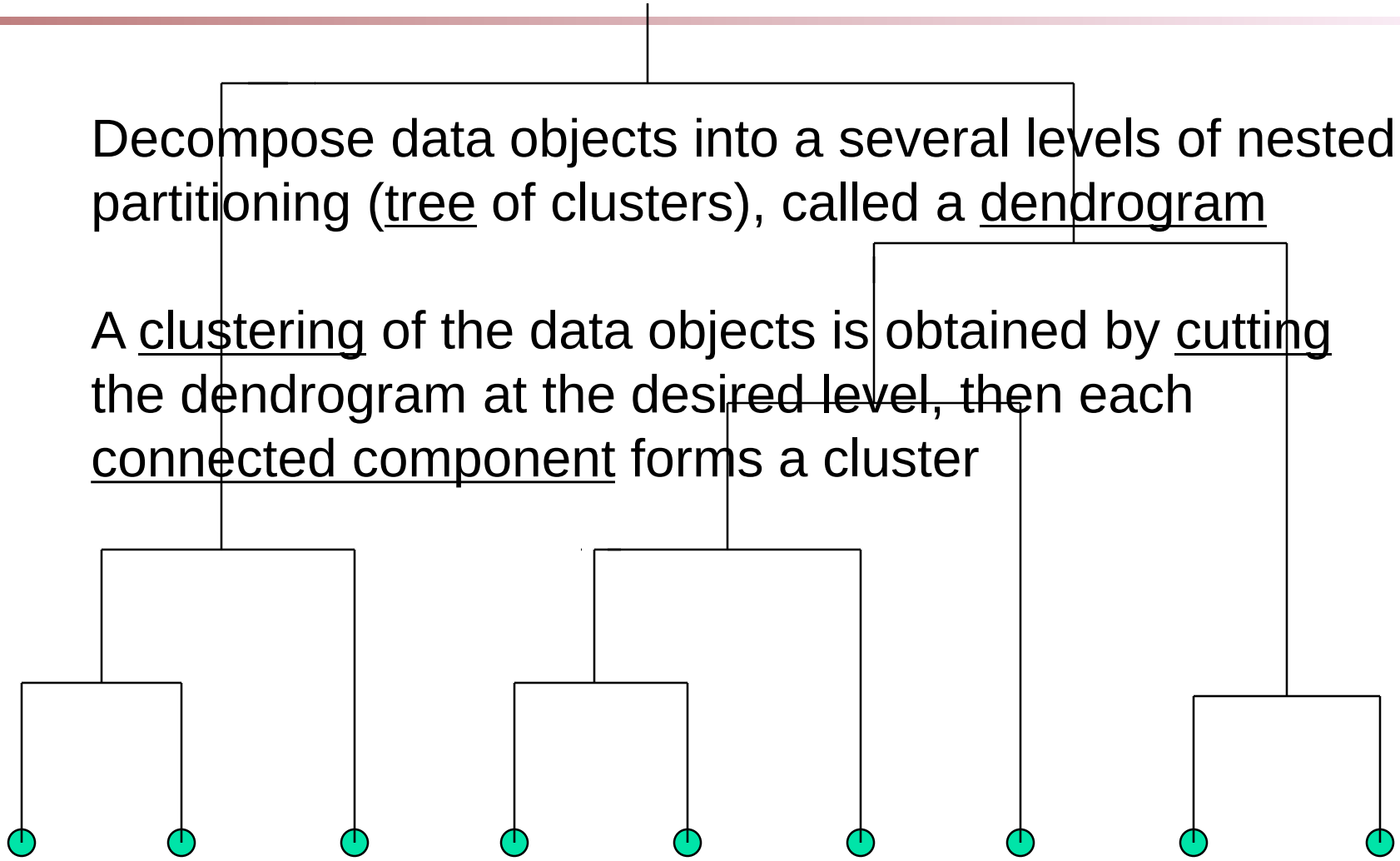
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

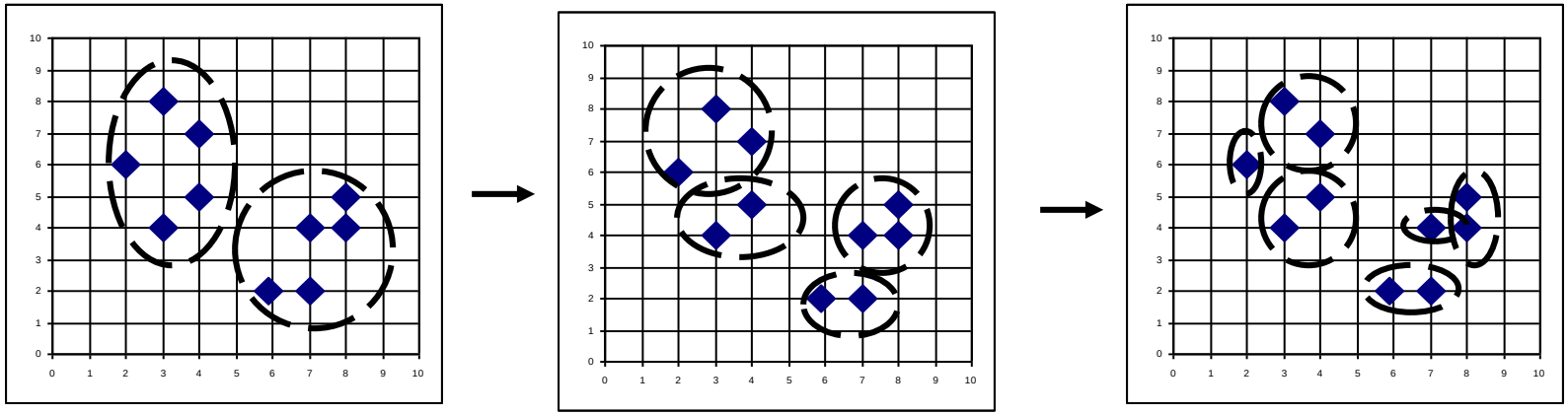
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

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

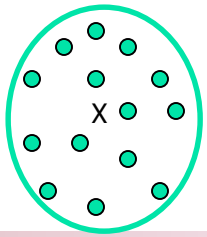


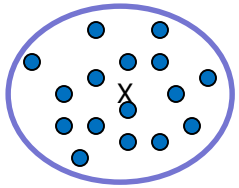
DIANA (Divisive Analysis)

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

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

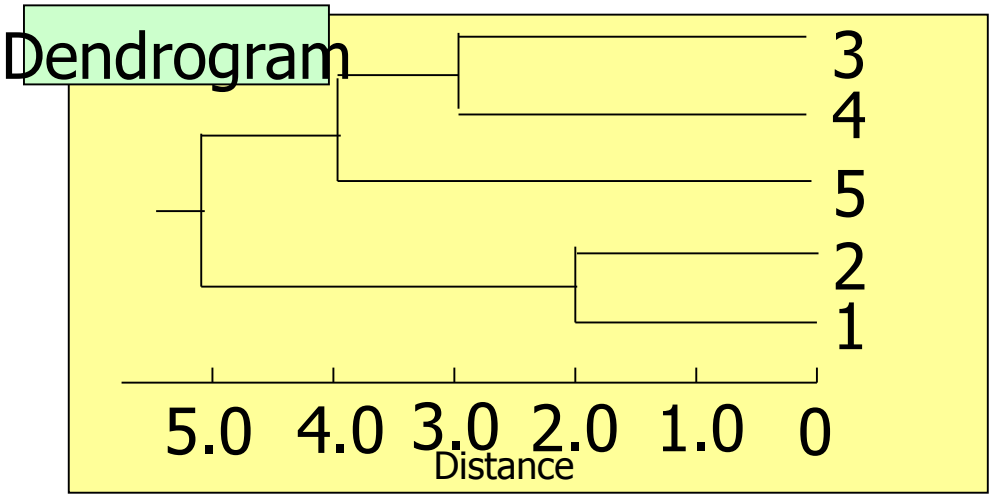
$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Dendrogram

- Hierarchic grouping can be represented by two-dimensional diagram known as a **dendrogram**.

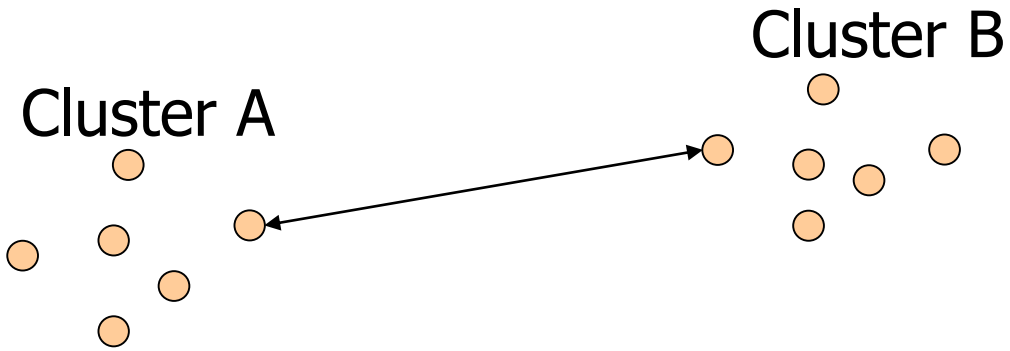


Distance

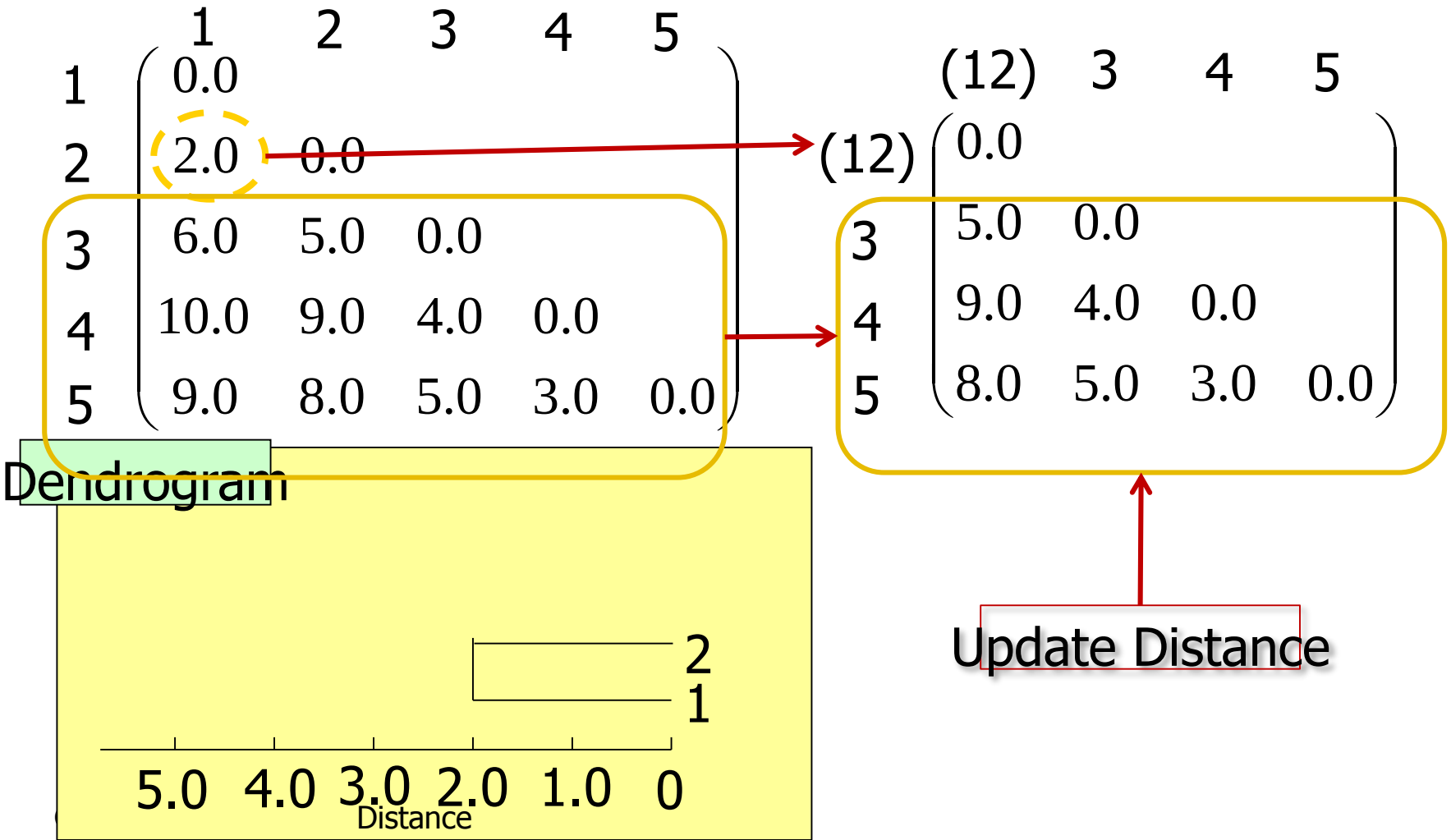
- Single Linkage
- Complete Linkage
- Group Average Linkage
- Centroid Linkage
- Median Linkage

Single Linkage

- Also, known as the **nearest neighbor** technique
- Distance between groups is defined as that of the closest pair of data, where only pairs consisting of one record from each group are considered

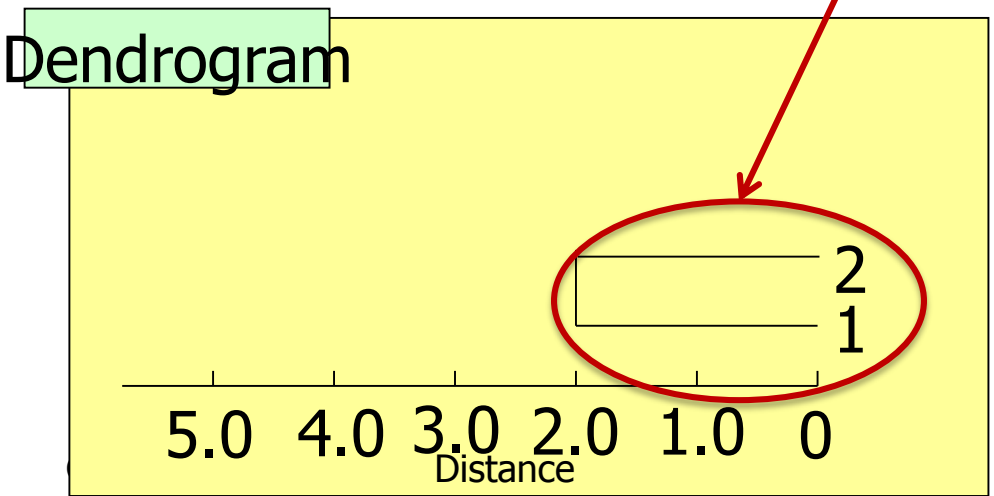


Find the smallest value, merge them



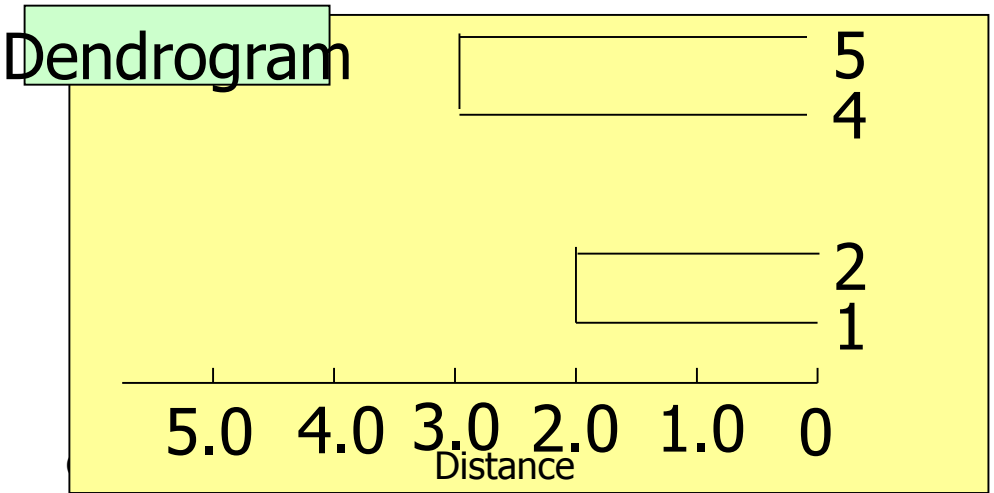
1 and 2 as a whole

	(12)	3	4	5
(12)	0.0			
3	5.0	0.0		
4	9.0	4.0	0.0	
5	8.0	5.0	3.0	0.0

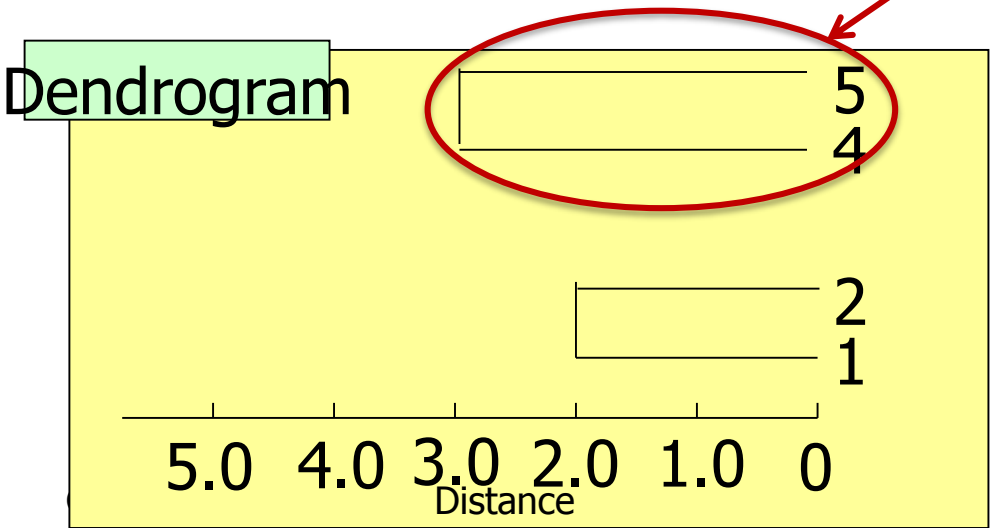


Repeat:
find the smallest one, merge

$$\begin{array}{c} \\ (12) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (12) & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.0 & & & \\ 5.0 & 0.0 & & \\ 9.0 & 4.0 & 0.0 & \\ 8.0 & 5.0 & 3.0 & 0.0 \end{pmatrix} \end{array} \rightarrow \begin{array}{c} \\ (12) \\ 3 \\ (4\ 5) \end{array}\begin{array}{c} \begin{array}{ccc} (12) & 3 & (4\ 5) \end{array} \\ \begin{pmatrix} 0.0 & & \\ 5.0 & 0.0 & \\ 8.0 & 4.0 & 0.0 \end{pmatrix} \end{array}$$

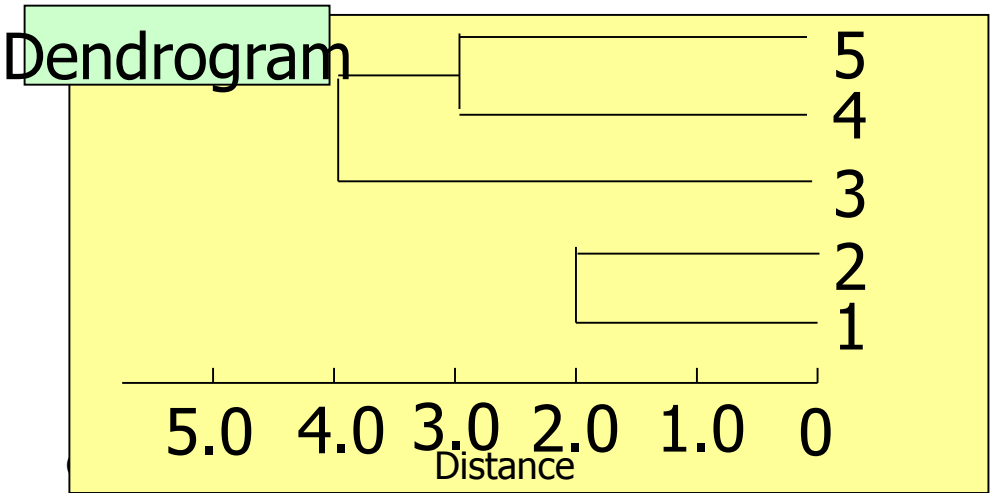


$$
\begin{array}{c} \\ (12) \\ 3 \\ (4\ 5) \end{array}
\begin{array}{c}
\begin{array}{ccc} (12) & 3 & (4\ 5) \end{array} \\
\begin{pmatrix} 0.0 & & \\ 5.0 & 0.0 & \\ 8.0 & 4.0 & 0.0 \end{pmatrix}
\end{array}
$$

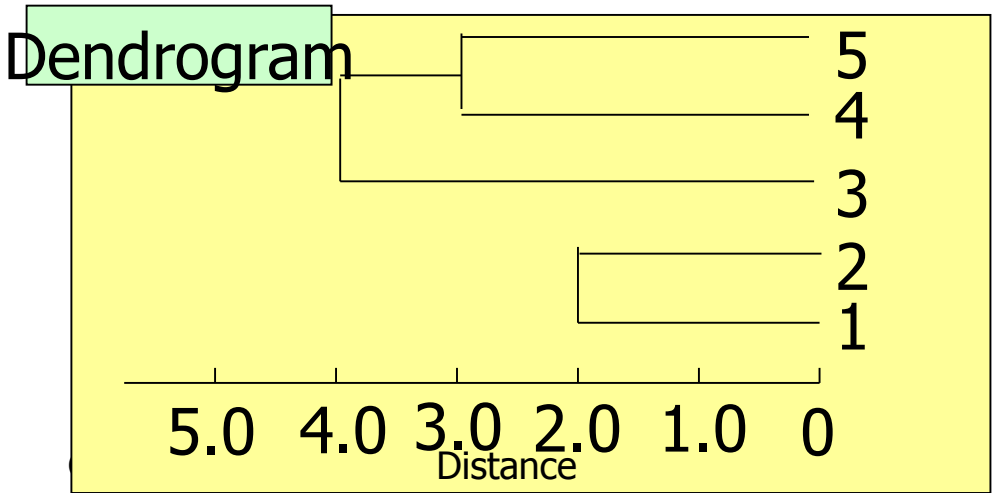


$$\begin{array}{c} \\ (12) \\ 3 \\ (4\ 5) \end{array} \begin{array}{c} \begin{array}{ccc} (12) & 3 & (4\ 5) \end{array} \\ \begin{pmatrix} 0.0 & & \\ 5.0 & 0.0 & \\ 8.0 & 4.0 & 0.0 \end{pmatrix} \end{array}$$

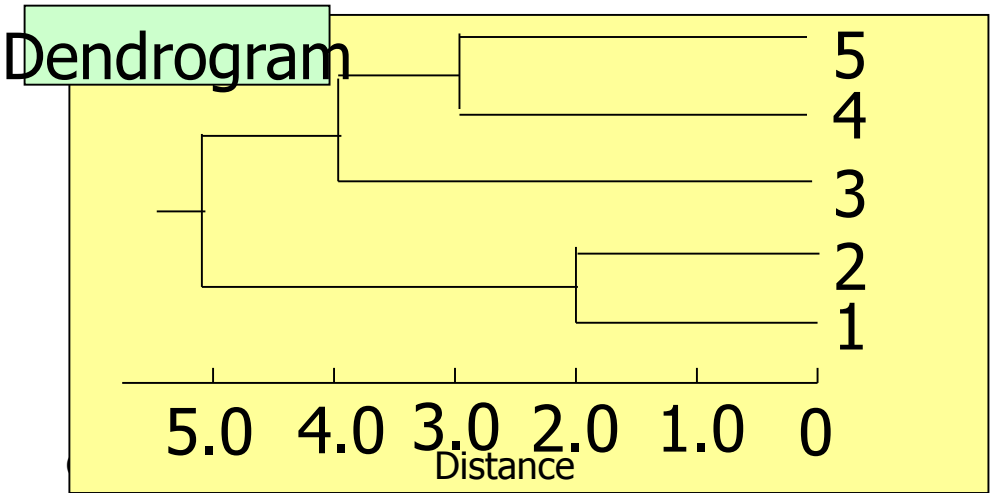
$$\begin{array}{c} \\ (12) \\ (3\ 4\ 5) \end{array} \begin{array}{c} \begin{array}{cc} (12) & (3\ 4\ 5) \end{array} \\ \begin{pmatrix} 0.0 & \\ 5.0 & 0.0 \end{pmatrix} \end{array}$$



$$
\begin{array}{c} \\ (12) \\ (3\ 4\ 5) \end{array}
\begin{array}{c}
\begin{array}{ccc} (12) & & (3\ 4\ 5) \end{array} \\
\begin{pmatrix} 0.0 & \\ 5.0 & 0.0 \end{pmatrix}
\end{array}
$$



$$
\begin{array}{c}
 & (12)(3\ 4\ 5) \\
(12) & \begin{pmatrix} 0.0 & \\ 5.0 & 0.0 \end{pmatrix} \\
(3\ 4\ 5) &
\end{array}
$$



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

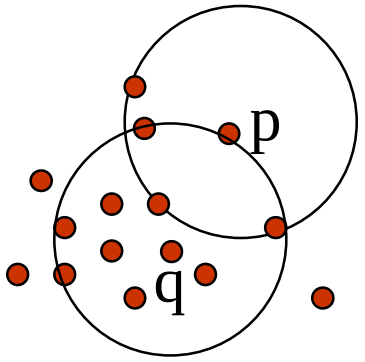
Density-Based Clustering Methods

- 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)

Density-Based Clustering: Basic Concepts

- Two parameters*:*
  - *Eps*: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood 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 $N_{Eps}(q)$
  - core point condition:

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

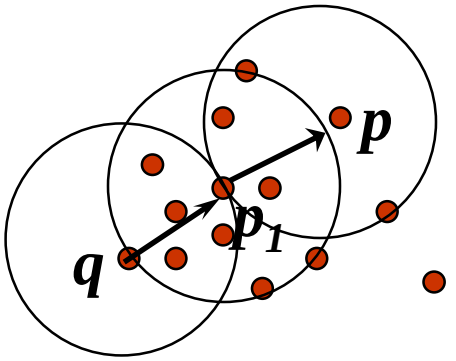


MinPts = 5

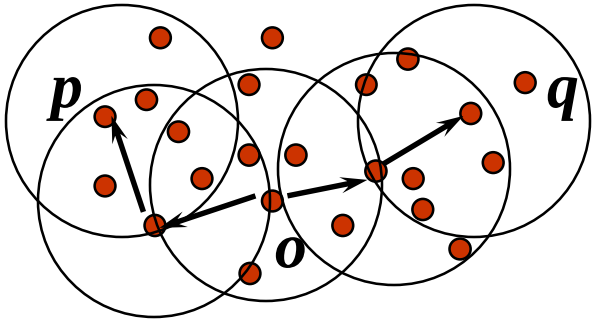
Eps = 1 cm

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, …, p_n, p_1 = *q*, p_n = *p* such that p_{i+1} is directly density-reachable from p_i

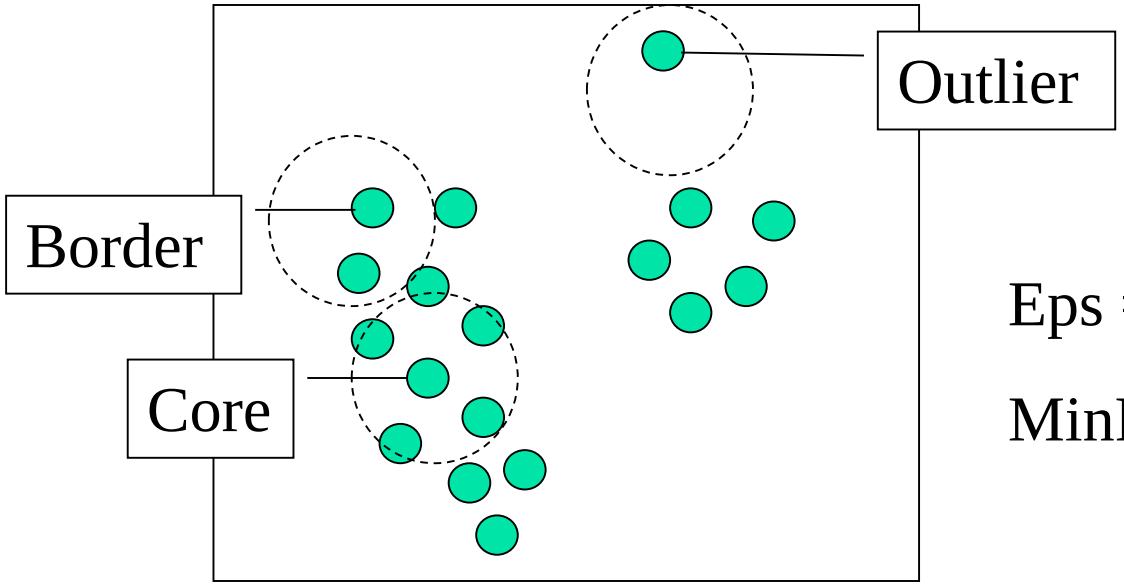


- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



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
- Discovers clusters of arbitrary shape in spatial databases with noise

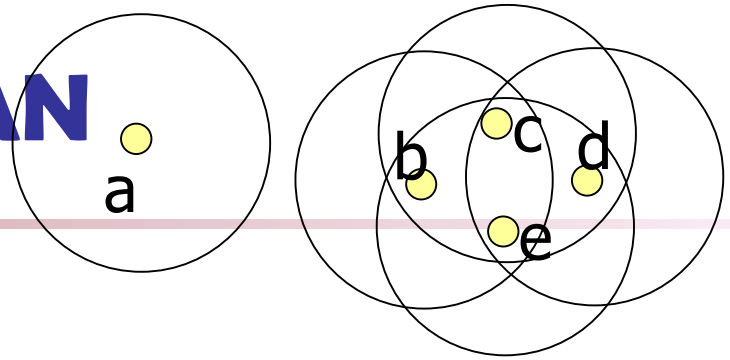


- A point is a core point if it has more than a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

DBSCAN: The Algorithm

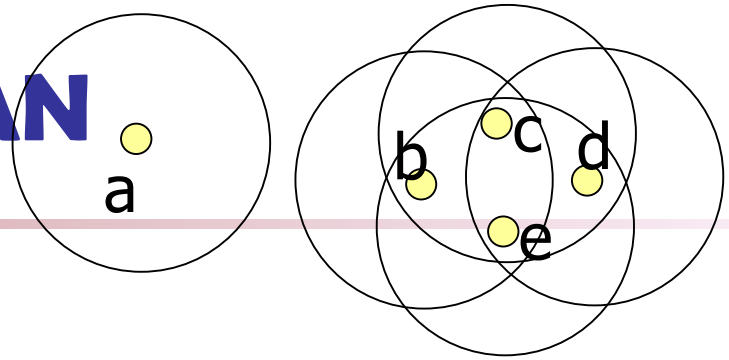
- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
- If *p* is a core point, a cluster is formed
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed
- *If a spatial index is used, the computational complexity of DBSCAN is O(nlogn), where n is the number of database objects. Otherwise, the complexity is* $O(n^2)$

DBSCAN



- Given a point p and a non-negative real number ε,
  - the ***ε-neighborhood*** of point p, denoted by N(p), is the set of points q (including point p itself) such that the distance between p and q is within ε.

DBSCAN



- According to ε-neighborhood of point p, we classify all points into three types
  - **core points**
  - **border points**
  - **noise points**

Given a point p and a non-negative integer MinPts, if the size of N(p) is at least MinPts, then p is said to be a **core point**.

Given a point p, p is said to be a **border point** if it is not a core point but N(p) contains at least one core point.

Given a point p, p is said to be a **noise point** if it is neither a core point nor a border point.

Example

- Consider the following 9 two-dimensional data points:

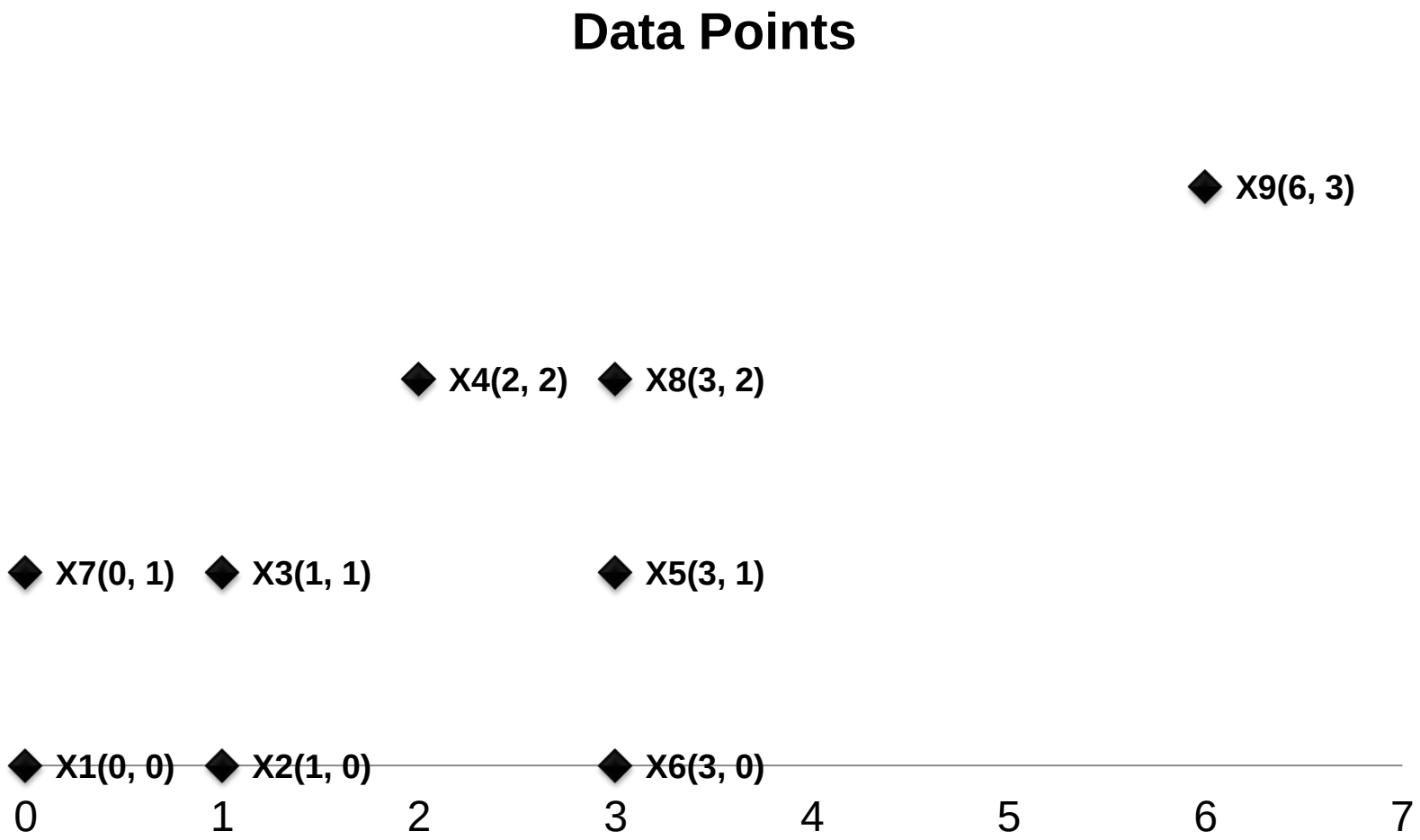
  x1(0,0), x2(1,0), x3(1,1), x4(2,2), x5(3,1), x6(3,0), x7(0,1), x8(3,2), x9(6,3)

Use the Euclidean Distance with Eps =1 and MinPts = 3

(Eps is short for epsilon: ε)

Find all core points, border points and noise points, and show the final clusters using DBCSAN algorithm.

Example


Data Points
X9(6, 3)
X4(2, 2)
X8(3, 2)
X7(0, 1)
X3(1, 1)
X5(3, 1)
X1(0, 0)
X2(1, 0)
X6(3, 0)
0
1
2
3
4
5
6
7

Calculate the N(p), *ε-neighborhood* of point p

- N(x1) = {x1, x2, x7}
- N(x2) = {x2, x1, x3}
- N(x3) = {x3, x2, x7}
- N(x4) = {x4, x8}
- N(x5) = {x5, x6, x8}
- N(x6) = {x6, x5}
- N(x7) = {x7, x1, x3}
- N(x8) = {x8, x4, x5}
- N(x9) = {x9}

Find all core points according to N(p)

- If the size of N(p) is at least MinPts, then p is said to be a **core point**.
- Here the given MinPts is 3, thus the size of N(p) is at least 3.
- We can find:
- N(x1) = {x1, x2, x7}
- N(x2) = {x2, x1, x3}
- N(x3) = {x3, x2, x7}
- N(x5) = {x5, x6, x8}
- N(x7) = {x7, x1, x3}
- N(x8) = {x8, x4, x5}

Find all core points according to N(p)

- Thus core points are:

  {x1, x2, x3, x5, x7, x8}

- Then according to the definition of border points: given a point p, p is said to be a **border point** if it is not a core point but N(p) contains at least one core point.

  N(x4) = {x4, x8}

  N(x6) = {x6, x5} ,

  here **x8 and x5** are **core points**, So both x4 and x6 are border points.

  Obviously, **x9** is a **noise** point.

Core points, Border points, Noise points

- **Core Points** are:

  {x1, x2, x3, x5, x7, x8}

- **Border Points**:

  {x4, x6}

- **Noise Points**:

  {x9}

DBSCAN: The Algorithm

- **Arbitrary** select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
- If *p* is a **core** point, a **cluster** is formed
- If *p* is a **border** point, no points are density-reachable from *p* and DBSCAN **visits the next point** of the database
- Continue the process until all of the points have been processed

DBSCAN: Example step by step

- **Arbitrary** select a point *p,* ***now we choose x1***
- Retrieve all points density-reachable from x1*:*

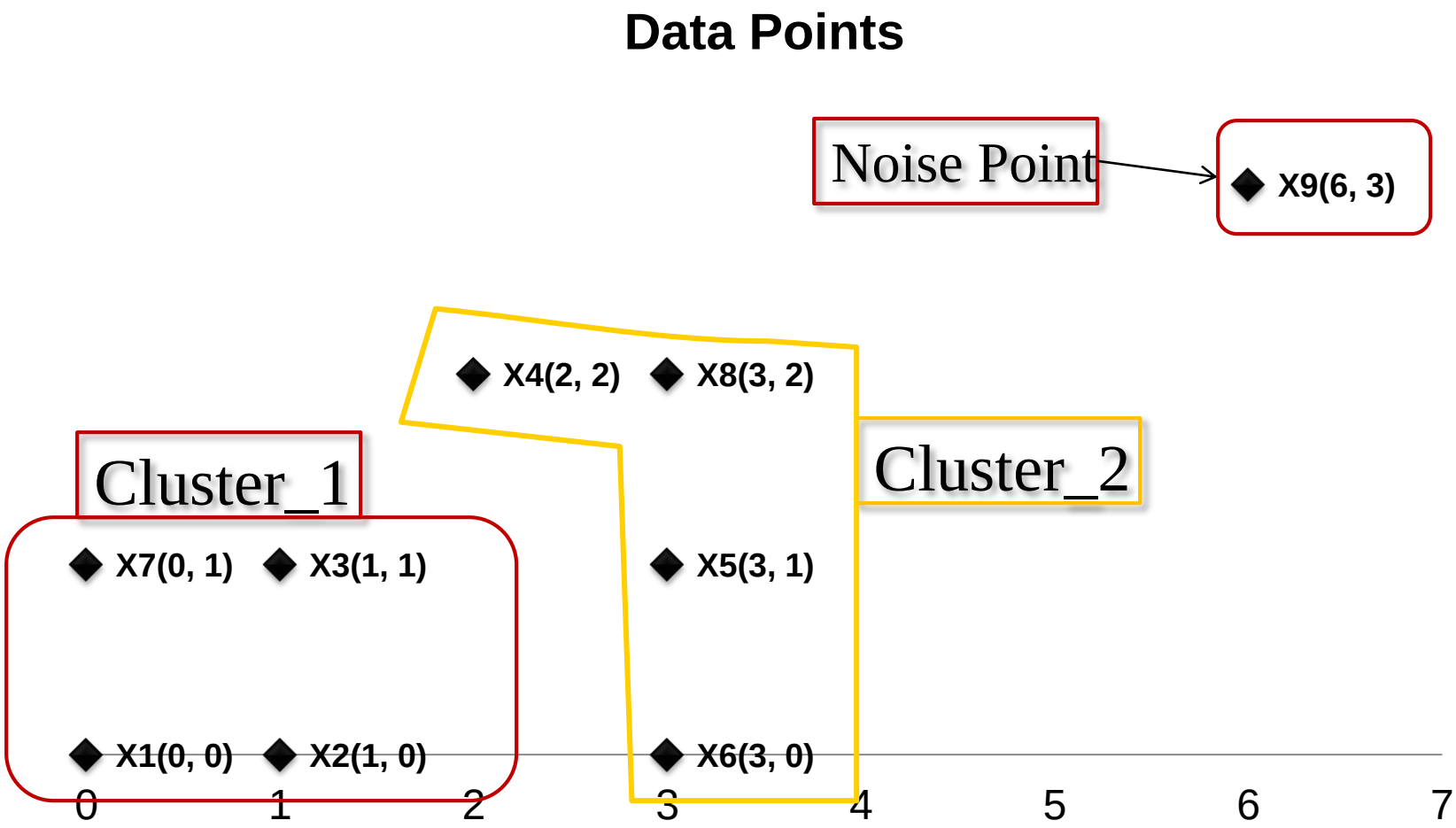
  {x2, x3, x7}
- Here x1 is a **core** point, a **cluster** is formed.
- So we have **Cluster_1: {x1, x2, x3, x7}**
- Next we choose **x5**,
- Retrieve all points density-reachable from x5*:*

  {x8, x4, x6}

DBSCAN: Example step by step

- Here x5 is a **core** point, a **cluster** is formed.
- So we have **Cluster_2: {x5, x4, x8, x6}**
- Next we choose **x9**, x9 is a noise point, noise points do NOT belong to any clusters.
- Thus the algorithm stops here.

Final Clusters using DBSCAN



DBSCAN: Sensitive to Parameters

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

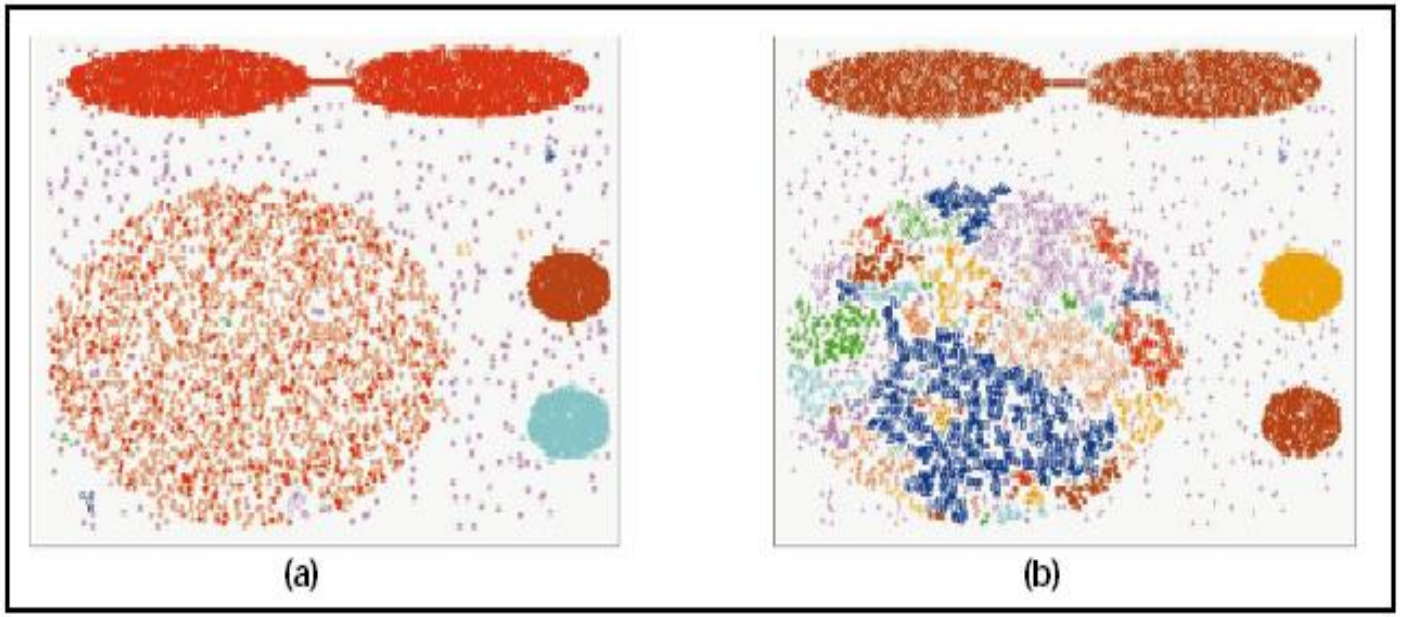
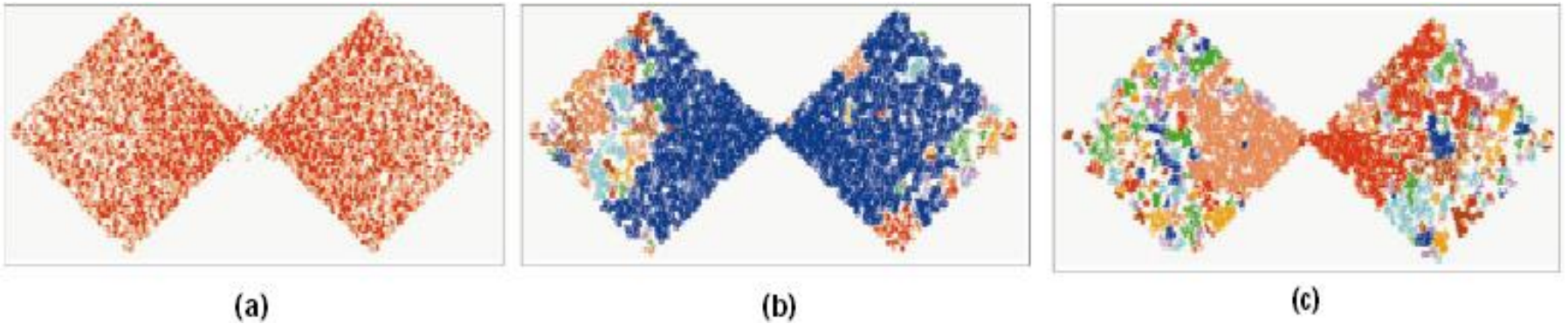


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN online Demo

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

Chapter 10. Cluster Analysis: Basic Concepts and Methods

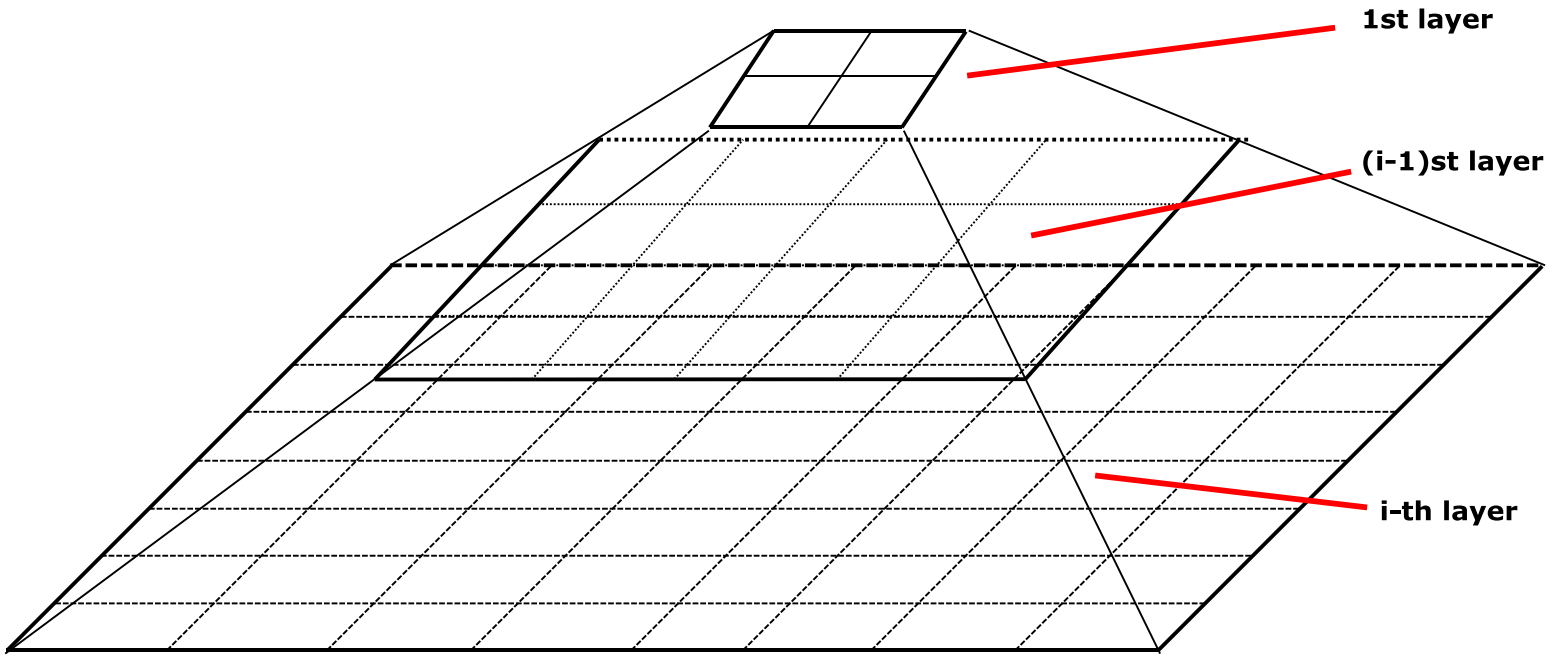
- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD'98)
    - Both grid-based and subspace clustering

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



The STING Clustering Method

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—*normal*, *uniform*, etc.
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

STING Algorithm and Its Analysis

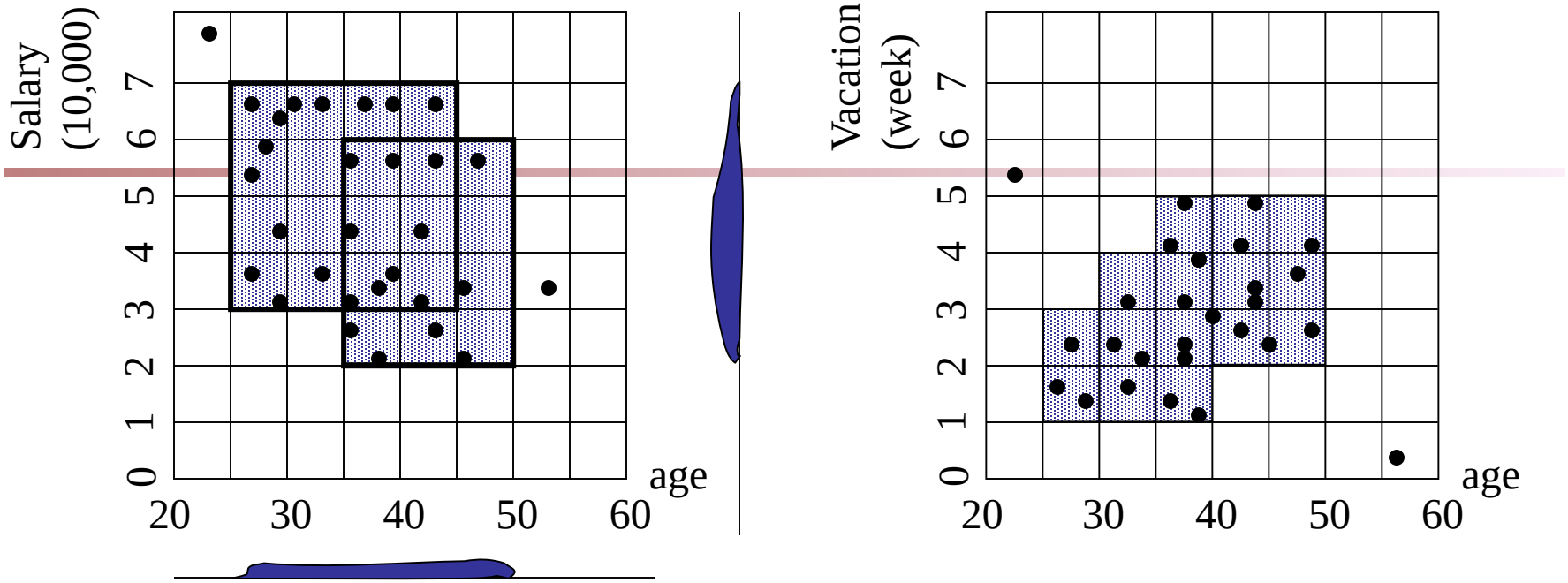
- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O(K)*, where *K* is the number of grid cells at the lowest level
- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

CLIQUE (Clustering In QUEst)

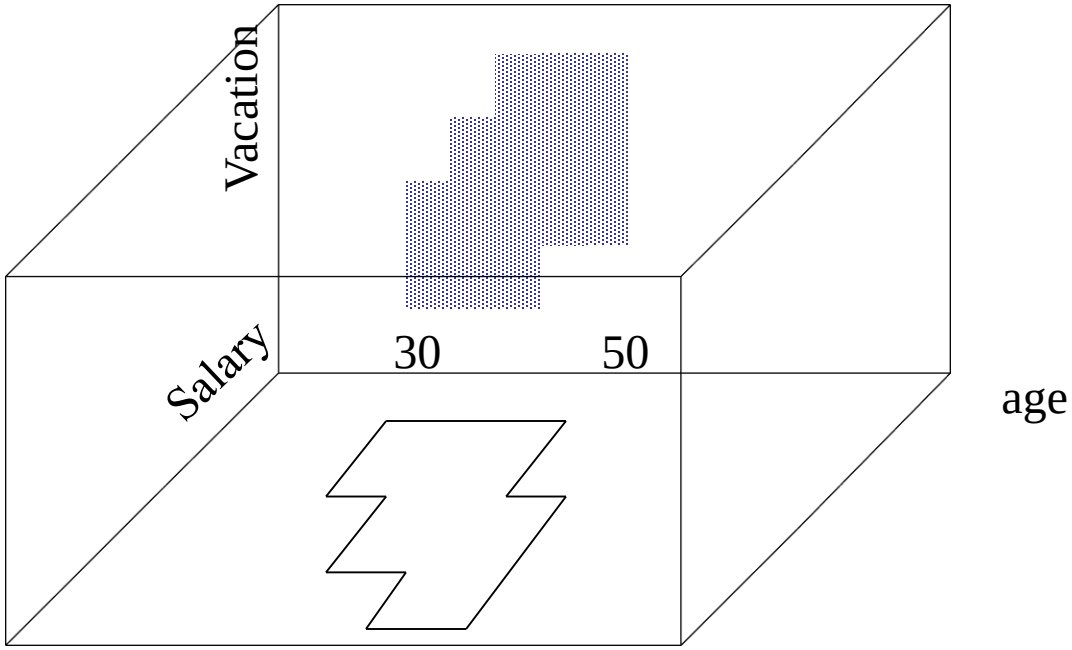
- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length interval
  - It partitions an m-dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster


Salary
(10,000)
0 1 2 3 4 5 6 7
20 30 40 50 60
age
Vacation
(week)
0 1 2 3 4 5 6 7
20 30 40 50 60
age

$\tau = 3$


Vacation
Salary
30
50
age

Strength and Weakness of *CLIQUE*

- Strength
  - *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - *insensitive* to the order of records in input and does not presume some canonical data distribution
  - scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

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 ≈√n/2 for a dataset of n points
- 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

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
  - Compare a clustering against the ground truth using certain clustering quality measure
  - Ex. BCubed precision and recall metrics
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
  - Ex. Silhouette coefficient

Sihouette coefficient

- For a data set, D, of n objects, suppose D is partitioned into k clusters, C_1, …$C_{K,}$ For each object $o \in D,$ we calculate:

a(o) as the average distance between o and all other object in the cluster to which o belongs.

b(o) is the minimum average distance from o to all clusters to which o does not belong.

$$a(o) = \frac{\sum_{o' \in Ci\ o \neq o'} dist(o, o')}{|C_i| - 1}$$

$$b(o) = min_{C_j:1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{o' \in Cj} dist(o, o')}{|Cj|} \right\}$$

Sihouette coefficient of o is:

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

Measuring Clustering Quality: Extrinsic Methods

- Clustering quality measure: $Q(C, C_g)$, for a clustering C given the ground truth C_g.
- 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

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

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.l 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. Bkasford. 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.