Machine Learning

TON DUC THANG UNIVERSITY
Faculty of Information Technology
Computer Science

Clustering

LÊ ANH CƯỜNG

Content

- 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)

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
  - 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)

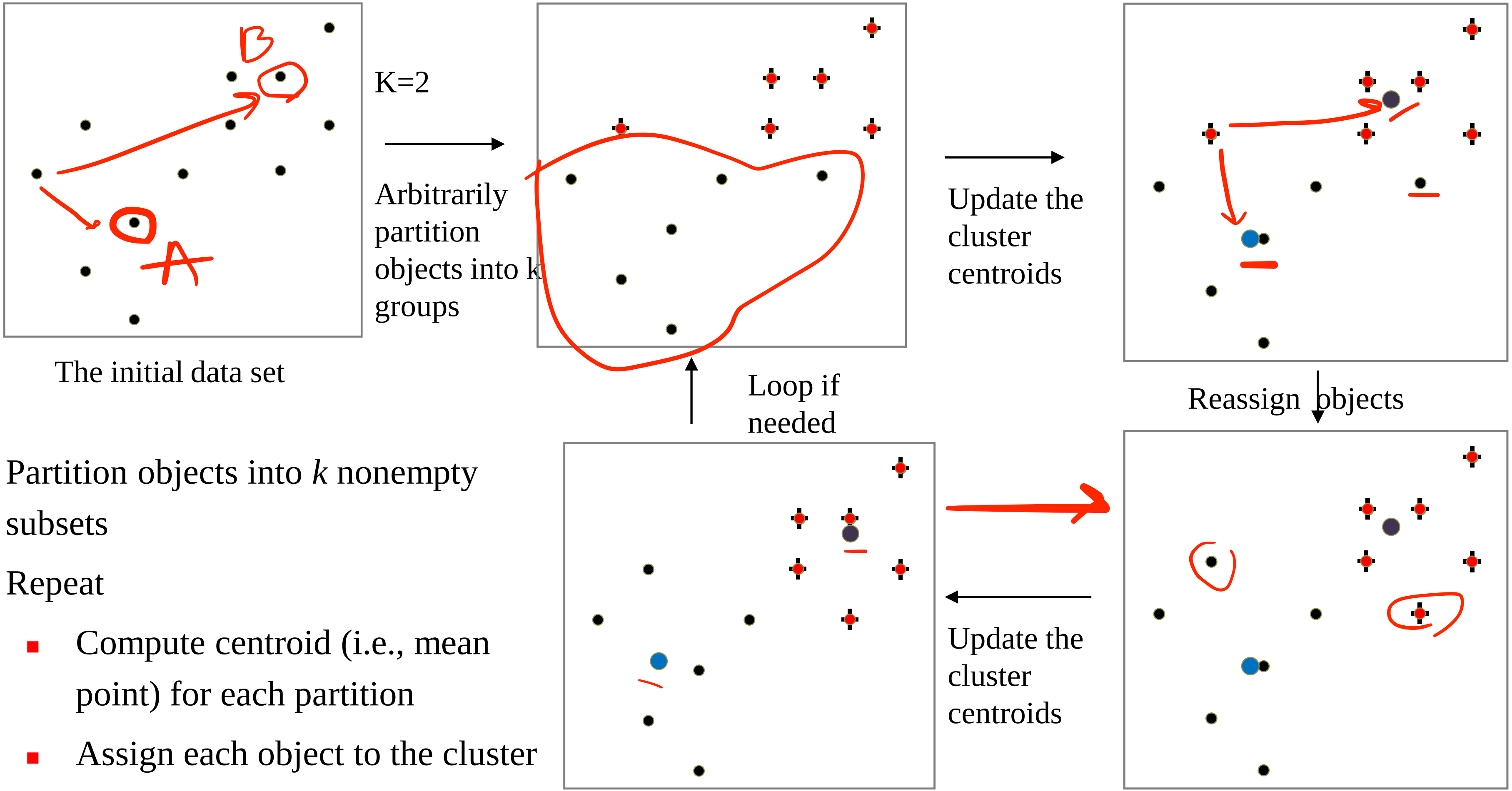
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- 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



The initial data set

K=2

Arbitrarily partition objects into k groups

Update the cluster centroids

Reassign objects

Loop if needed

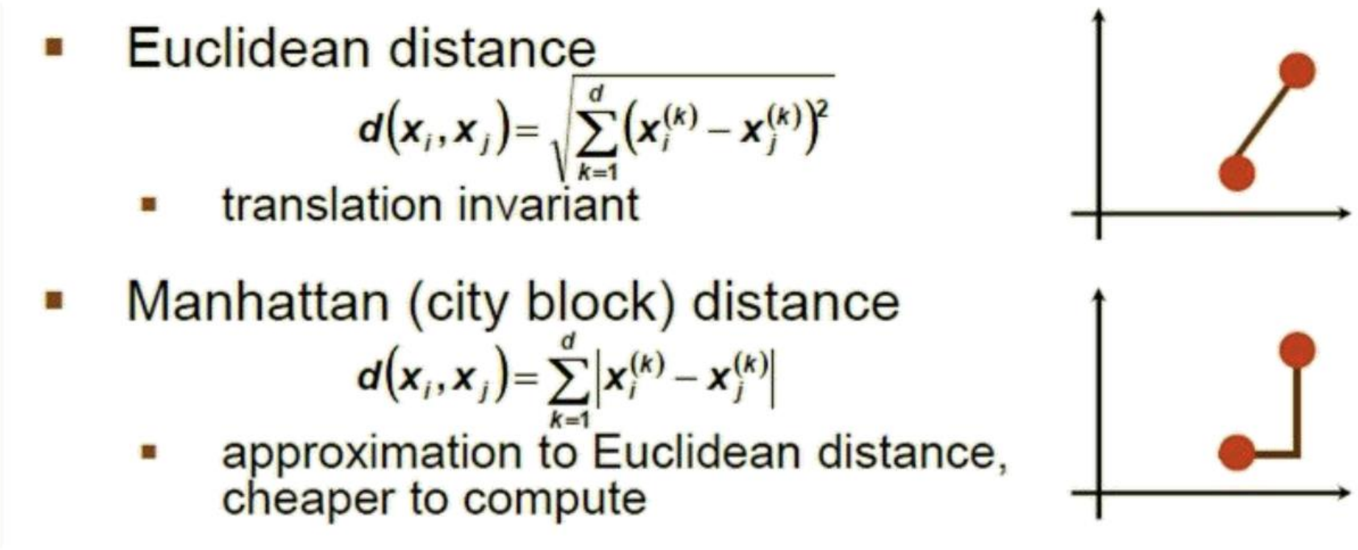
Update the cluster centroids

- 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

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
    - Sensitive to noisy data and *outliers*
    - Not suitable to discover clusters with *non-convex shapes*

Distance (dissimilarity) measures



- Euclidean distance

  $$d(\mathbf{x}_i,\mathbf{x}_j)=\sqrt{\sum_{k=1}^{d}\left(x_i^{(k)}-x_j^{(k)}\right)^2}$$

  - translation invariant

- Manhattan (city block) distance

  $$d(\mathbf{x}_i,\mathbf{x}_j)=\sum_{k=1}^{d}\left|x_i^{(k)}-x_j^{(k)}\right|$$

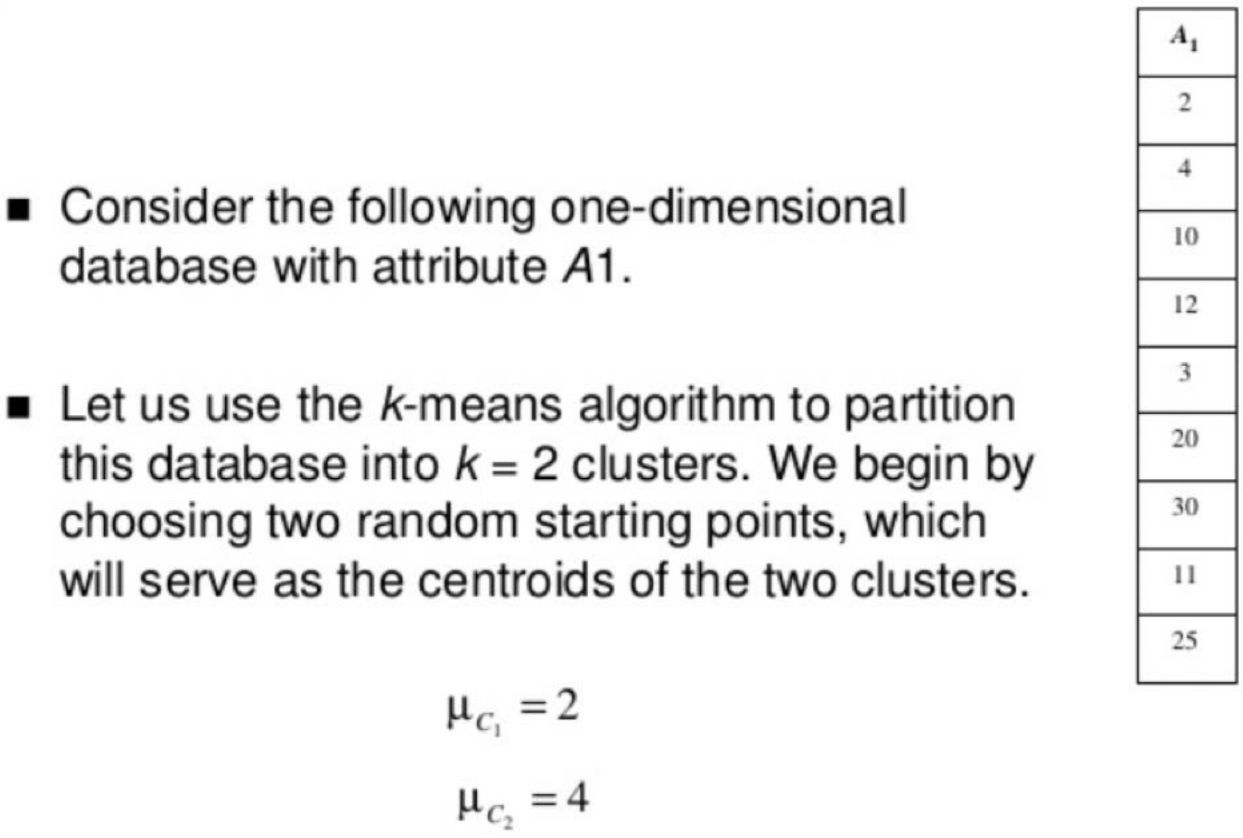
  - approximation to Euclidean distance, cheaper to compute

- They are special cases of **Minkowski distance**:

$$d_p(\mathbf{x}_i,\mathbf{x}_j)=\left(\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right|^p\right)^{\frac{1}{p}}$$

(p is a positive integer)

Example



- Consider the following one-dimensional database with attribute *A*1.
- Let us use the *k*-means algorithm to partition this database into *k* = 2 clusters. We begin by choosing two random starting points, which will serve as the centroids of the two clusters.

$$\mu_{C_1} = 2$$

$$\mu_{C_2} = 4$$

A_1
2
4
10
12
3
20
30
11
25

$C_1 = (2, 3)$

$C_2 = (4, 10, 12, 20, 30, 11, 25)$

- Consider the following one-dimensional database with attribute *A*1.

- Let us use the *k*-means algorithm to partition this database into *k* = 2 clusters. We begin by choosing two random starting points, which will serve as the centroids of the two clusters.

$$\mu_{C_1} = 2$$

$$\mu_{C_2} = 4$$

A_1
2
4
10
12
3
20
30
11
25

$C_1 = (2, 3) \rightarrow 2.5$

$C_2 = (4, 10, 12, 20, 30, 11, 25) \rightarrow 16$

- Consider the following one-dimensional database with attribute *A*1.

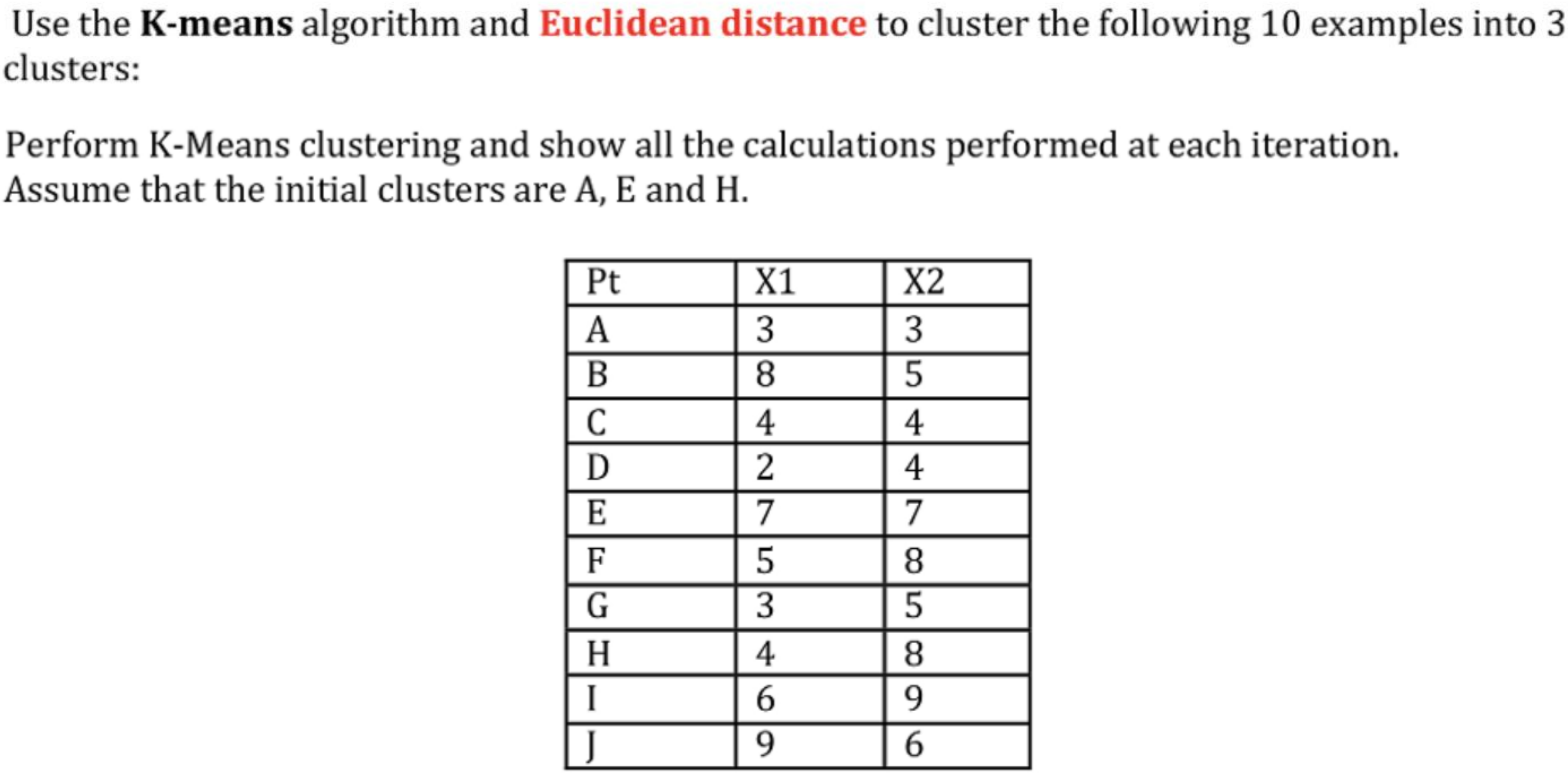
- Let us use the *k*-means algorithm to partition this database into *k* = 2 clusters. We begin by choosing two random starting points, which will serve as the centroids of the two clusters.

A_1	
2	1
4	—
10	—
12	—
3	—
20	—
30	—
11	—
25	—

$$\mu_{C_1} = 2$$

$$\mu_{C_2} = 4$$

Exercise



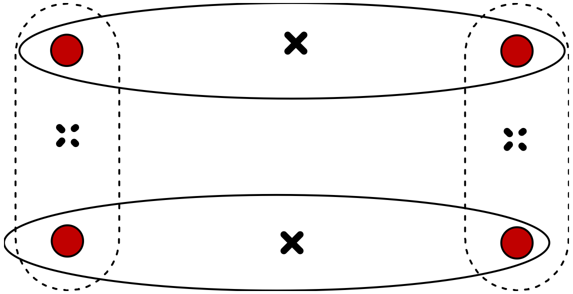
Use the **K-means** algorithm and **Euclidean distance** to cluster the following 10 examples into 3 clusters:

Perform K-Means clustering and show all the calculations performed at each iteration. Assume that the initial clusters are A, E and H.

Pt	X1	X2
A	3	3
B	8	5
C	4	4
D	2	4
E	7	7
F	5	8
G	3	5
H	4	8
I	6	9
J	9	6

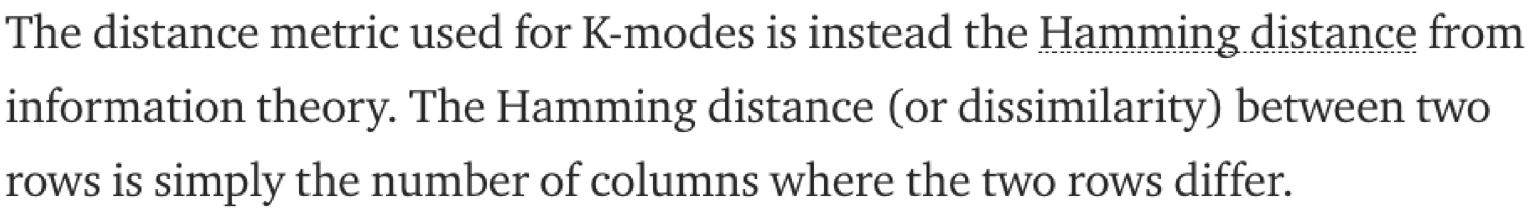
Variations of the *K-Means* Method

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

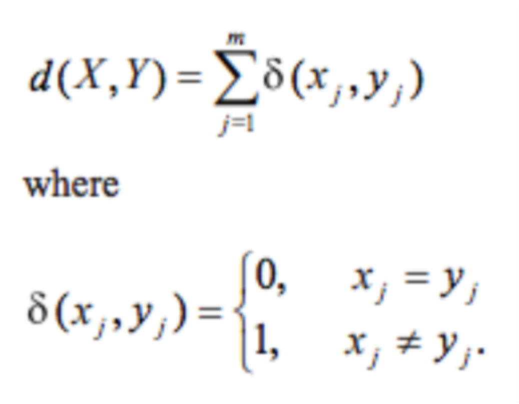


- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

K-modes



The distance metric used for K-modes is instead the Hamming distance from information theory. The Hamming distance (or dissimilarity) between two rows is simply the number of columns where the two rows differ.



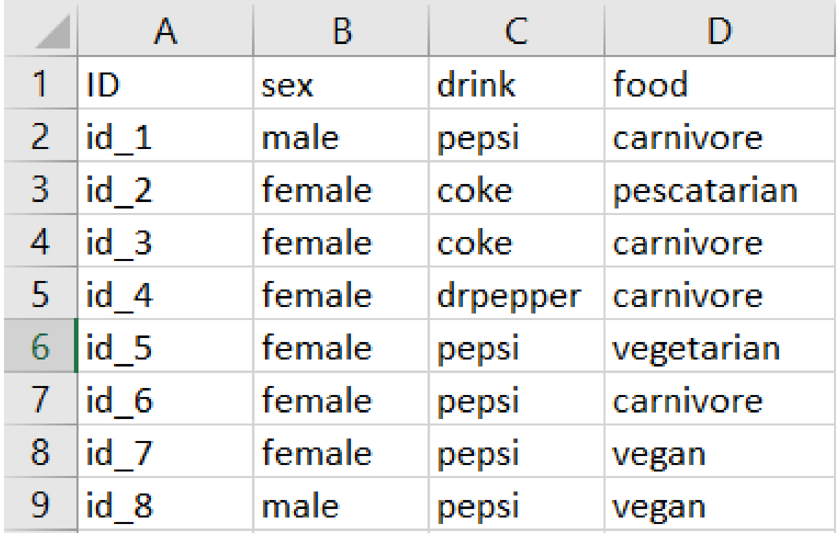
$$d(X,Y)=\sum_{j=1}^{m}\delta(x_j,y_j)$$

where

$$\delta(x_j,y_j)=\begin{cases}0, & x_j=y_j\\ 1, & x_j\neq y_j.\end{cases}$$

Exercise

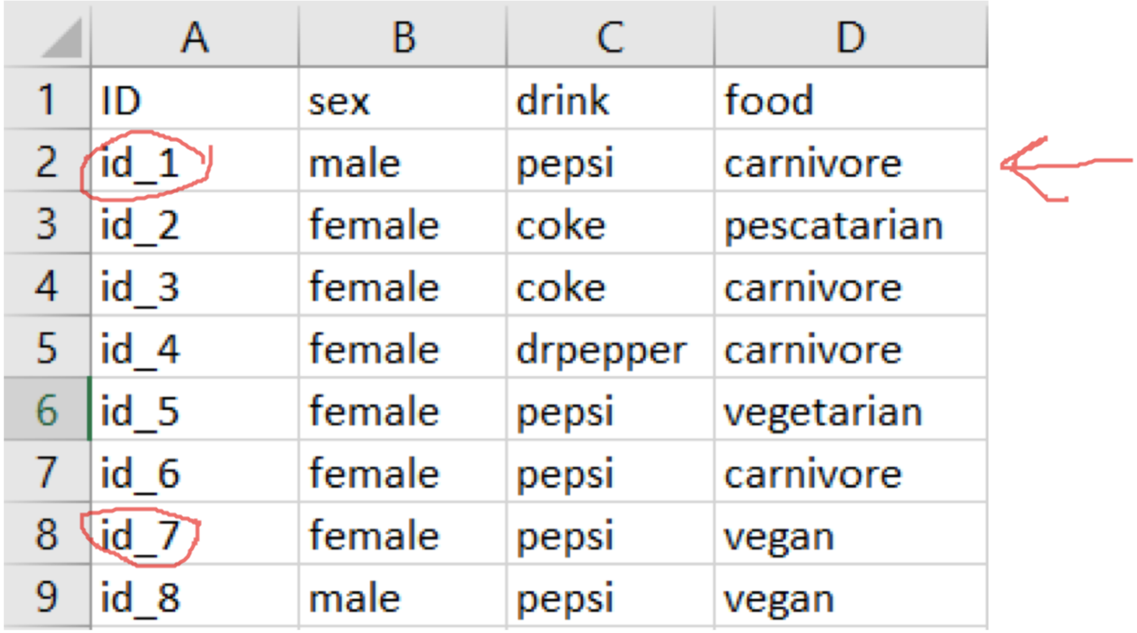
Cluster these elements into 2 clusters



	A	B	C	D
1	ID	sex	drink	food
2	id_1	male	pepsi	carnivore
3	id_2	female	coke	pescatarian
4	id_3	female	coke	carnivore
5	id_4	female	drpepper	carnivore
6	id_5	female	pepsi	vegetarian
7	id_6	female	pepsi	carnivore
8	id_7	female	pepsi	vegan
9	id_8	male	pepsi	vegan

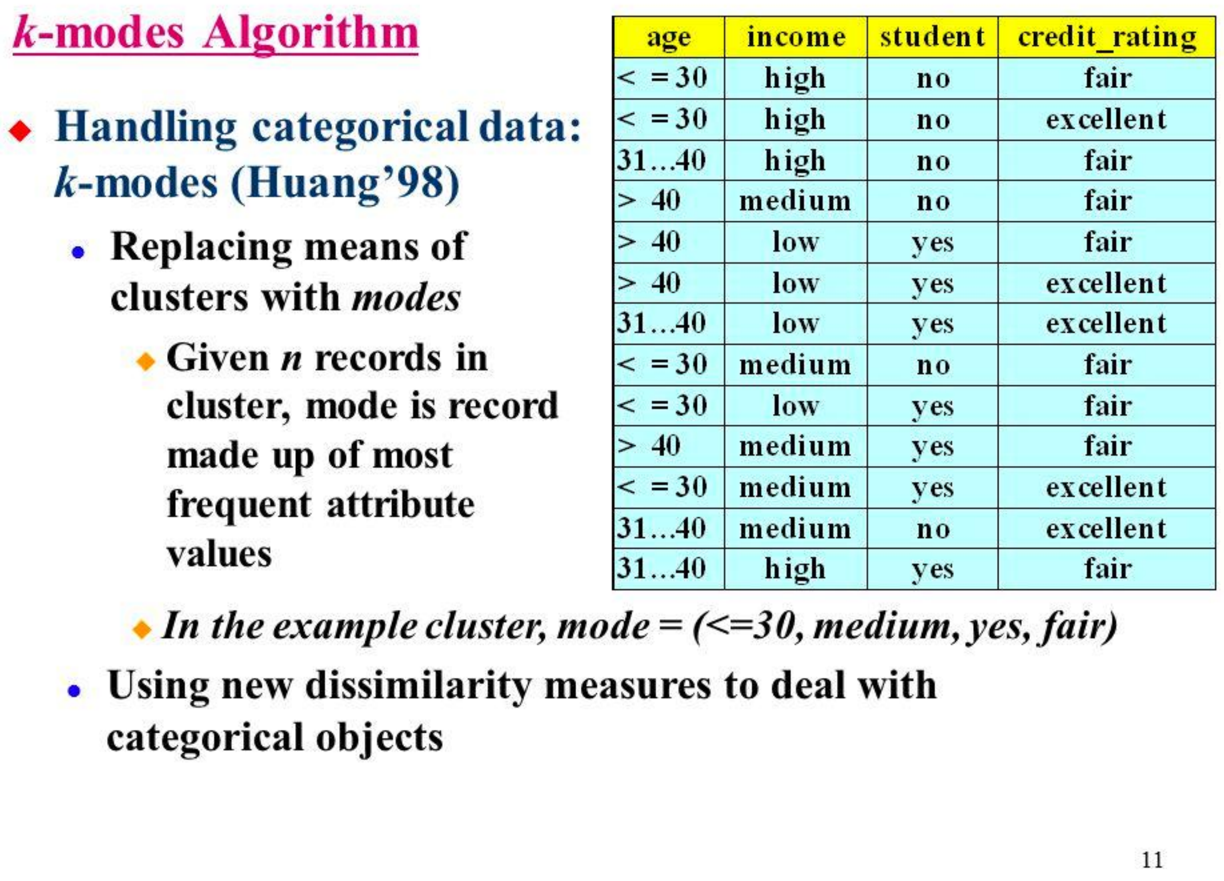
K-modes

Cluster these elments into 2 clusters



	A	B	C	D
1	ID	sex	drink	food
2	id_1	male	pepsi	carnivore
3	id_2	female	coke	pescatarian
4	id_3	female	coke	carnivore
5	id_4	female	drpepper	carnivore
6	id_5	female	pepsi	vegetarian
7	id_6	female	pepsi	carnivore
8	id_7	female	pepsi	vegan
9	id_8	male	pepsi	vegan

K-modes



k-modes Algorithm

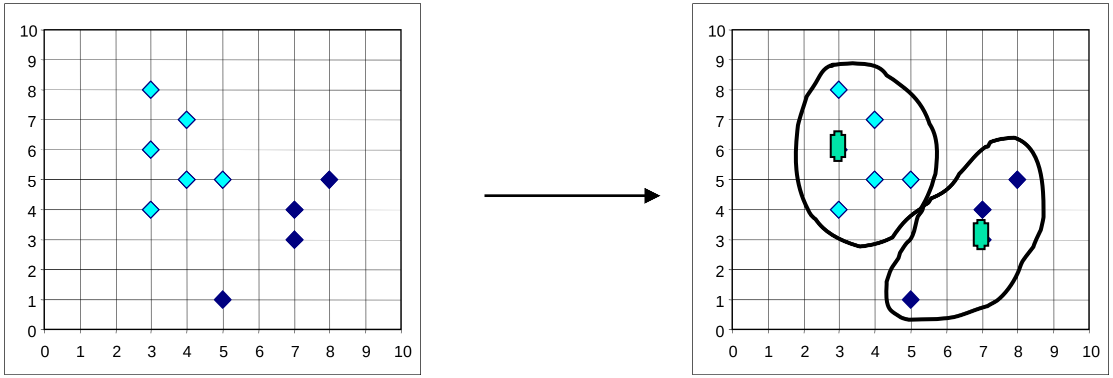
- Handling categorical data: *k*-modes (Huang'98)
  - Replacing means of clusters with *modes*
    - Given *n* records in cluster, mode is record made up of most frequent attribute values
    - *In the example cluster, mode = (<=30, medium, yes, fair)*
  - Using new dissimilarity measures to deal with categorical objects

age	income	student	credit_rating
< =30	high	no	fair
< =30	high	no	excellent
31…40	high	no	fair
> 40	medium	no	fair
> 40	low	yes	fair
> 40	low	yes	excellent
31…40	low	yes	excellent
< =30	medium	no	fair
< =30	low	yes	fair
> 40	medium	yes	fair
< =30	medium	yes	excellent
31…40	medium	no	excellent
31…40	high	yes	fair

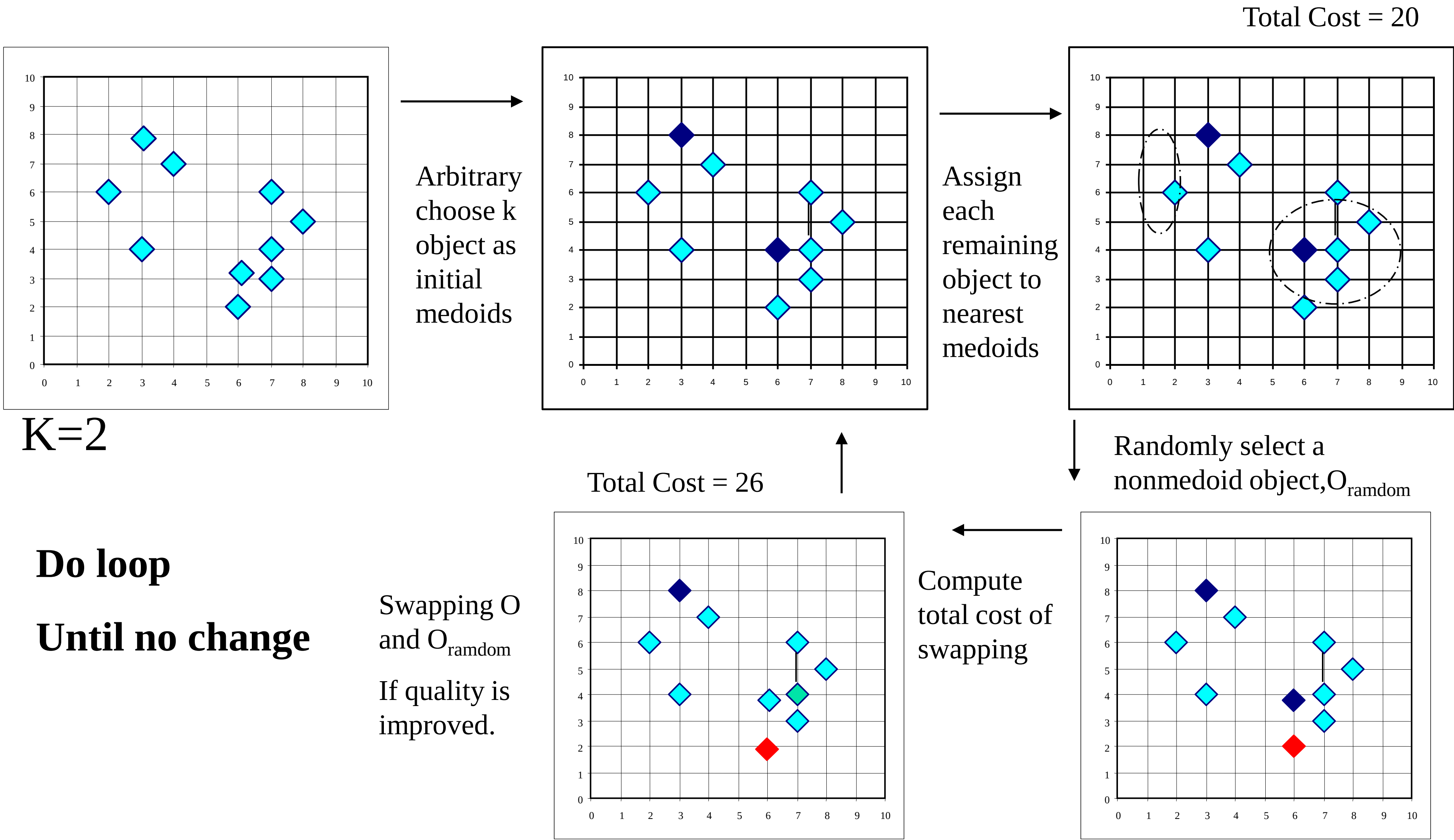
11

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

Total Cost = 20

Assign each remaining object to nearest medoids

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

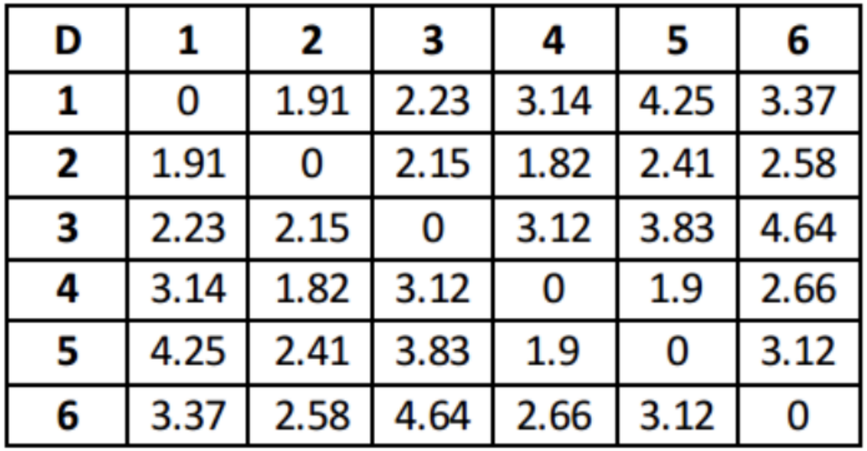
The K-Medoid Clustering Method

- *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

Example

k=2

$C_1 = \{X_1, X_2, X_3\}, C_2 = \{X_4, X_5, X_6\}$



D	1	2	3	4	5	6
1	0	1.91	2.23	3.14	4.25	3.37
2	1.91	0	2.15	1.82	2.41	2.58
3	2.23	2.15	0	3.12	3.83	4.64
4	3.14	1.82	3.12	0	1.9	2.66
5	4.25	2.41	3.83	1.9	0	3.12
6	3.37	2.58	4.64	2.66	3.12	0

1. Set K to the desired number of clusters, lets use 2.
2. Choose randomly K entities to be the medoids m_1, m_2. Lets choose X_3 (Lets call this cluster 1) and X_5 (Cluster 2).
3. Assign a given entity to the cluster represented by its closest medoid. Cluster 1 will be made of entities (X_1, X_2, X_3 - just check your table, these are closer to X_3 than to X_5), cluster 2 will be (X_4, X_5, X_6).
4. Update the medoids. A medoid of a cluster should be the entity with the smallest sum of distances to all other entities within the same cluster. X_2 will be the new medoid for cluster 1, and X_4 for cluster 2.

Example

X_2

k=2

$C_1 = \{X_1, X_2, X_3\}, C_2 = \{X_4, X_5, X_6\}$

D	1	2	3	4	5	6
1	0	1.91	2.23	3.14	4.25	3.37
2	1.91	0	2.15	1.82	2.41	2.58
3	2.23	2.15	0	3.12	3.83	4.64
4	3.14	1.82	3.12	0	1.9	2.66
5	4.25	2.41	3.83	1.9	0	3.12
6	3.37	2.58	4.64	2.66	3.12	0

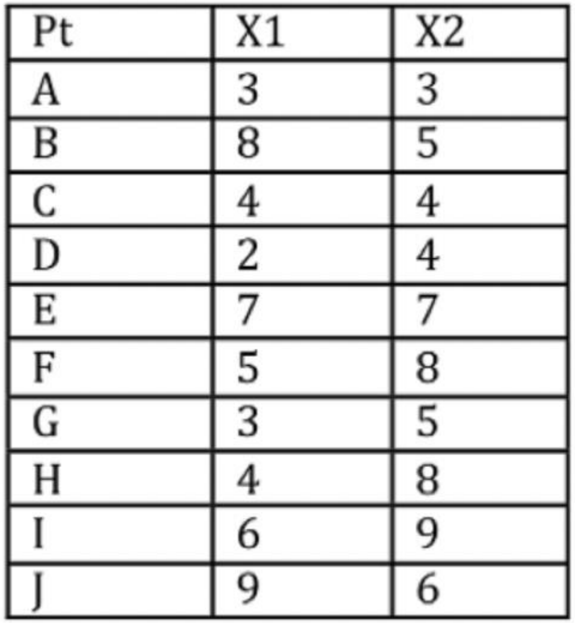
4.38
4.06
4.18

1. Set K to the desired number of clusters, lets use 2.
2. Choose randomly K entities to be the medoids m_1, m_2. Lets choose X_3 (Lets call this cluster 1) and X_5 (Cluster 2).
3. Assign a given entity to the cluster represented by its closest medoid. Cluster 1 will be made of entities (X_1, X_2, X_3 - just check your table, these are closer to X_3 than to X_5), cluster 2 will be (X_4, X_5, X_6).
4. Update the medoids. A medoid of a cluster should be the entity with the smallest sum of distances to all other entities within the same cluster. X_2 will be the new medoid for cluster 1, and X_4 for cluster 2.

Exercise

Use the **K-means** algorithm and **Euclidean distance** to cluster the following 10 examples into 3 clusters:

Perform K-Means clustering and show all the calculations performed at each iteration.
Assume that the initial clusters are A, E and H.



Pt	X1	X2
A	3	3
B	8	5
C	4	4
D	2	4
E	7	7
F	5	8
G	3	5
H	4	8
I	6	9
J	9	6

Project

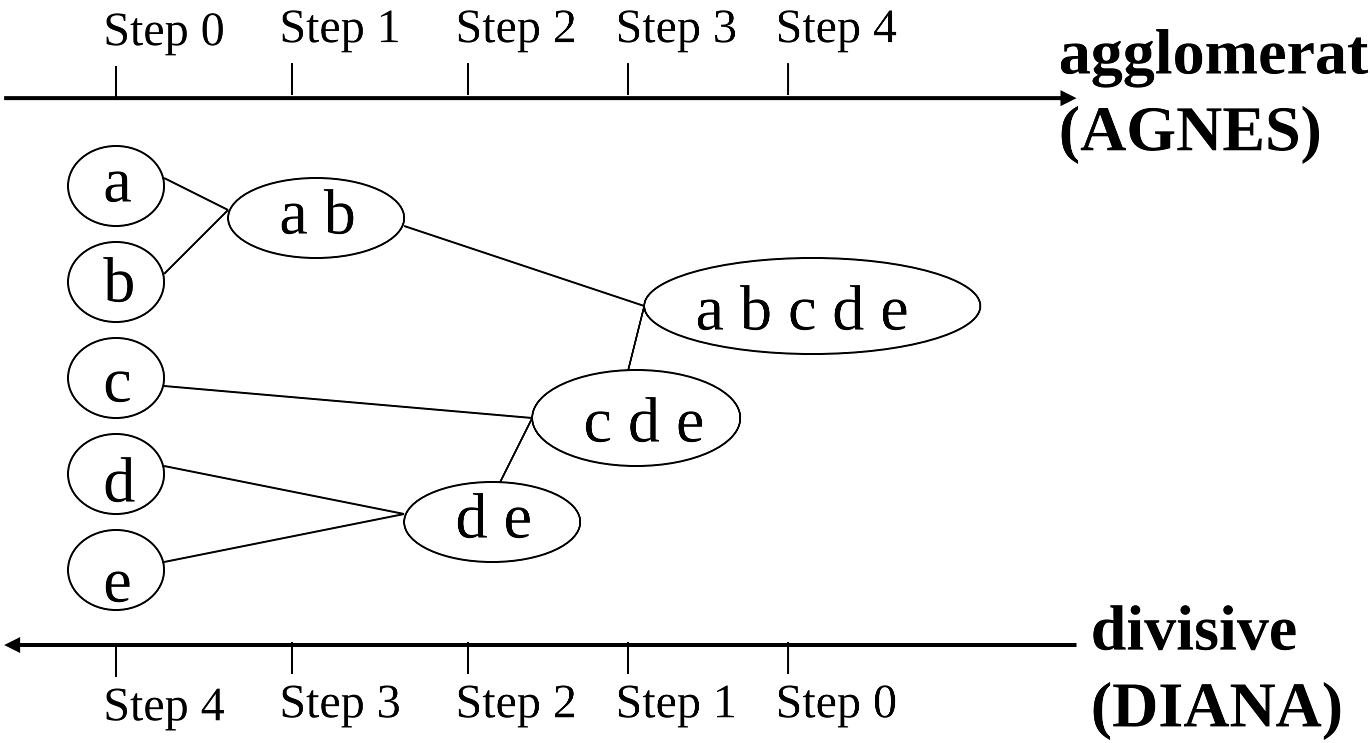
- Given a set of N sentences.
- Cluster these N sentences into k clusters.

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

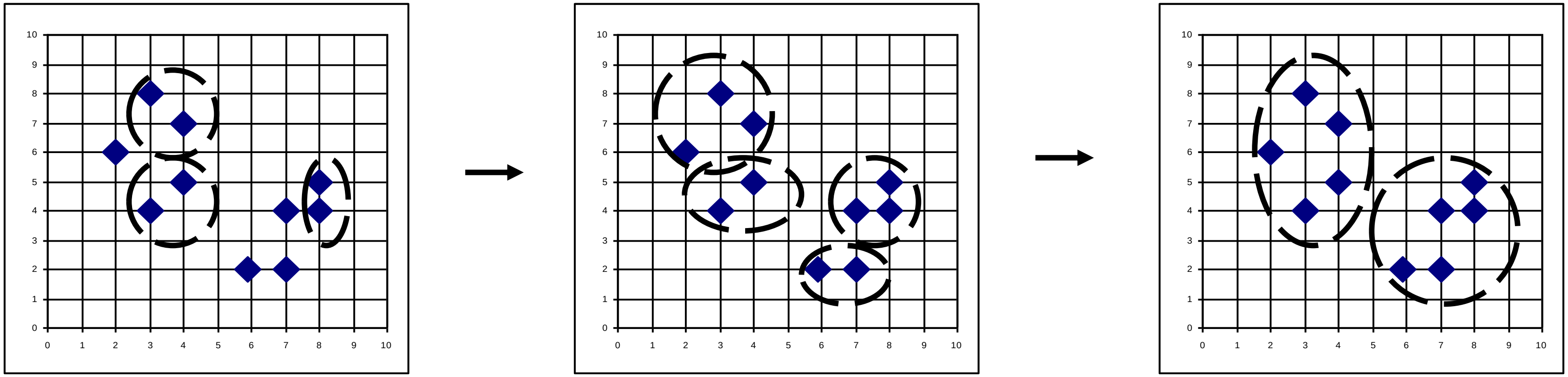
Hierarchical Clustering

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

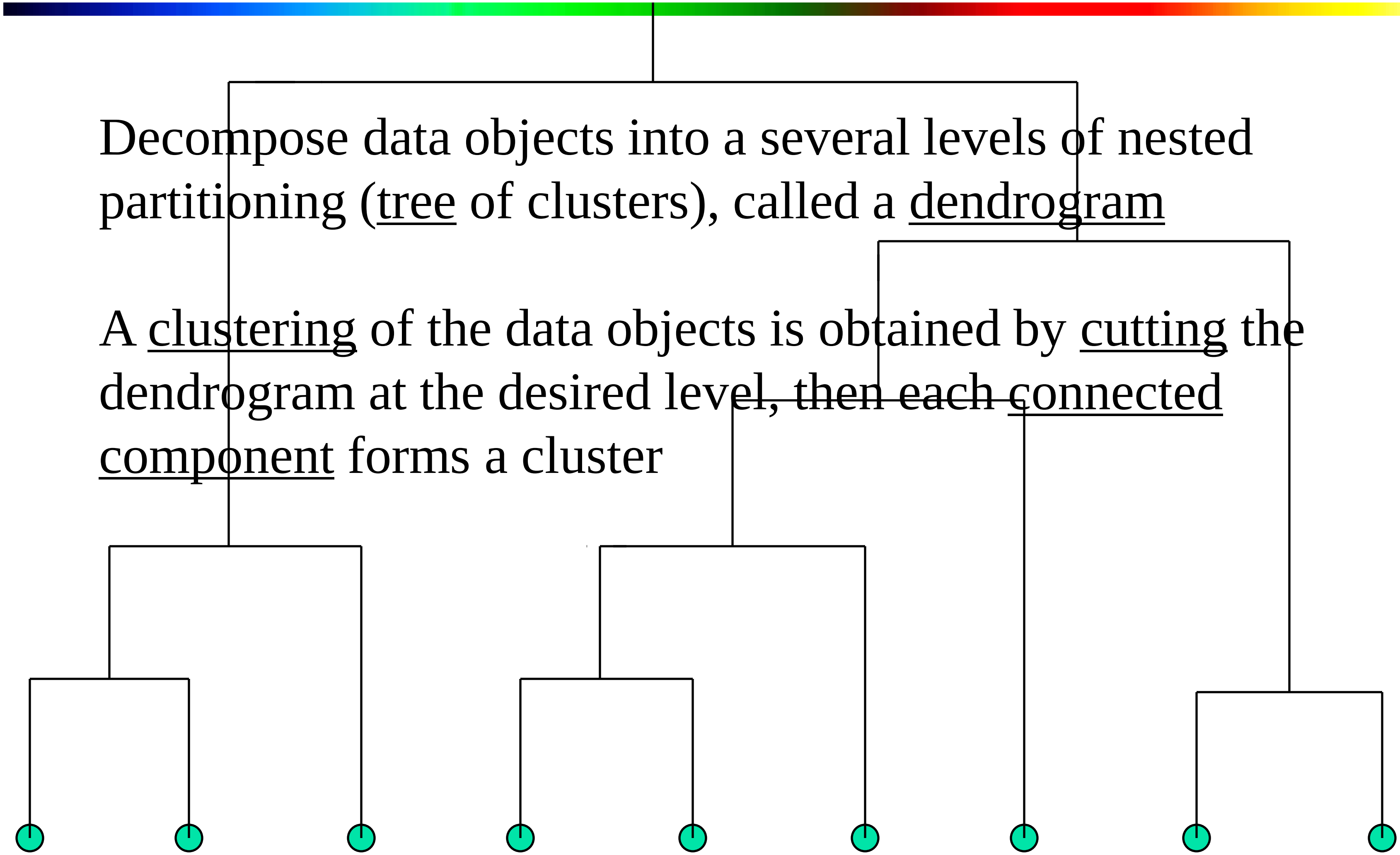


AGNES (Agglomerative Nesting)

- 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

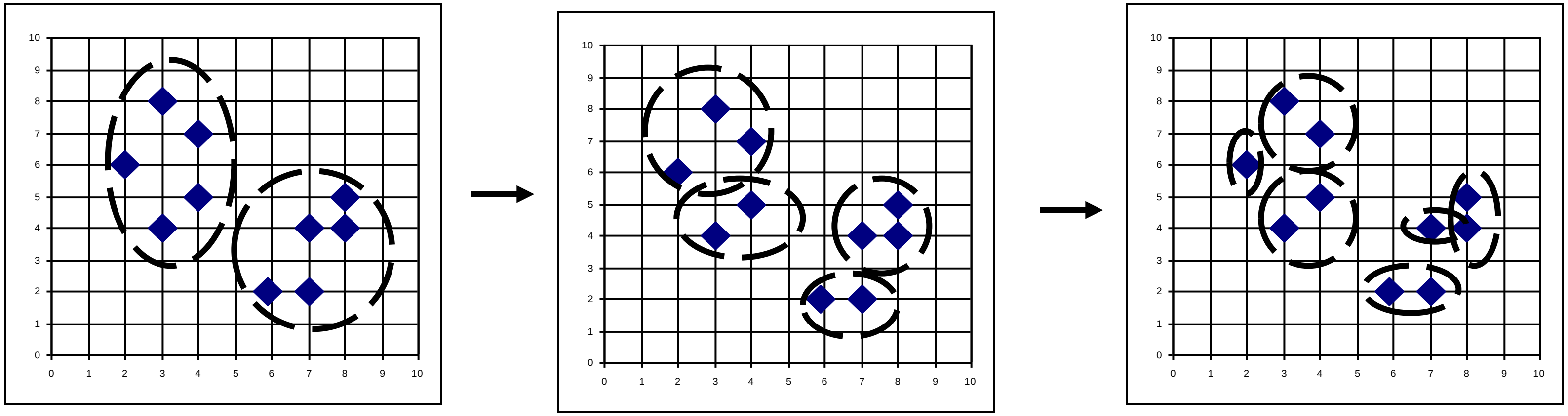


Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

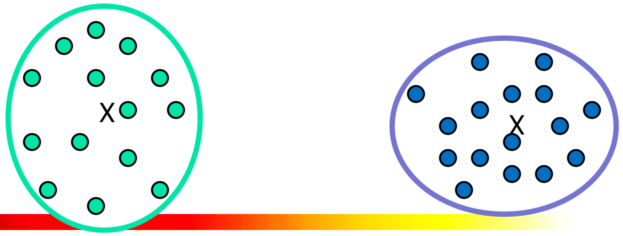
A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

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., $\text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- 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

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

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

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

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

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

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

Clustering Feature (CF): ***CF = (N, LS, SS)***

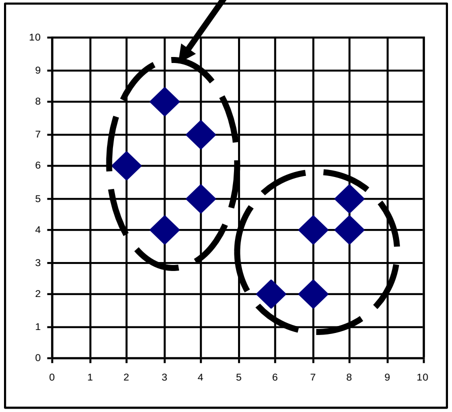
N: **Number of data points**

LS: linear sum of N points: $\sum_{i=1}^{N} X_i$

SS: square sum of N points

$$\sum_{i=1}^{N} X_i^{\,2}$$

CF = (5, (16,30),(54,190))



(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

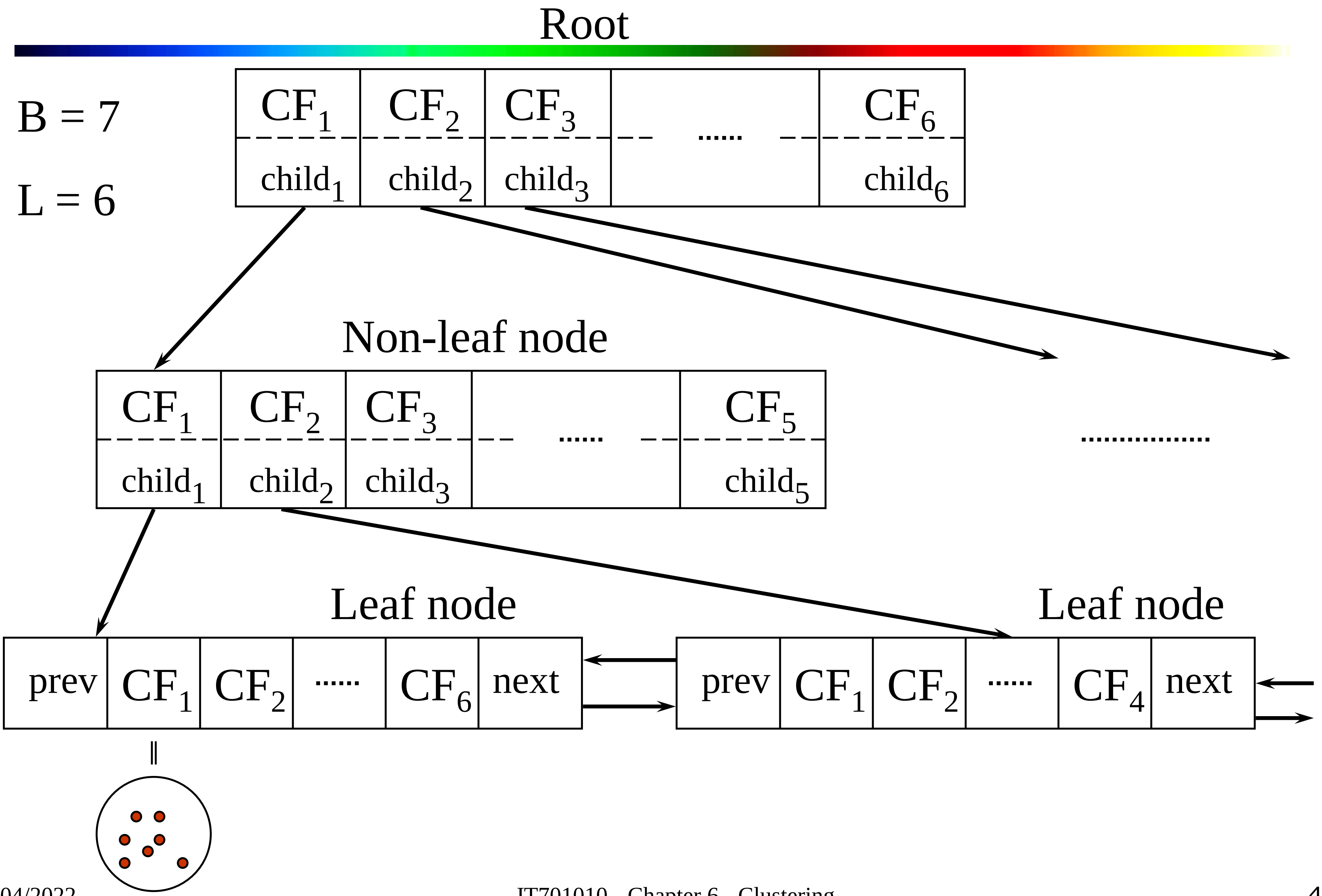
CF-Tree in BIRCH

- Clustering feature:
  - Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
  - Registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or “children”
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: max # of children
  - Threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure

B = 7

L = 6


Root
CF_1 CF_2 CF_3 ...... CF_6
$child_1$ $child_2$ $child_3$ $child_6$
Non-leaf node
CF_1 CF_2 CF_3 ...... CF_5
$child_1$ $child_2$ $child_3$ $child_5$
....
Leaf node
prev CF_1 CF_2 ...... CF_6 next
Leaf node
prev CF_1 CF_2 ...... CF_4 next
||

The Birch Algorithm

- Cluster Diameter

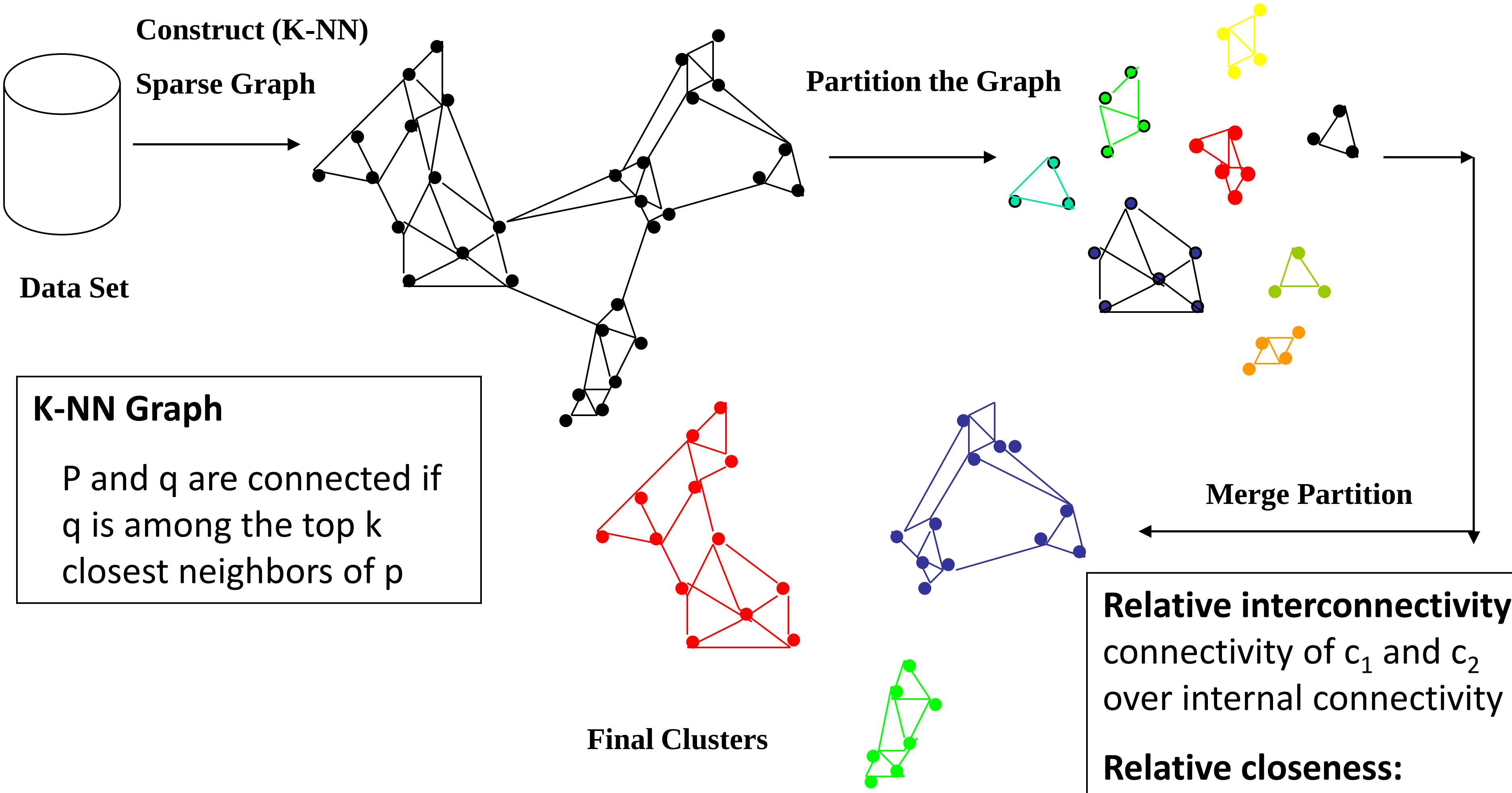
$$\sqrt{\frac{1}{n(n-1)}\sum(x_i - x_j)^2}$$

- For each point in the input
  - Find closest leaf entry
  - Add point to leaf entry and update CF
  - If entry diameter > max_diameter, then split leaf, and possibly parents
- Algorithm is O(n)
- Concerns
  - Sensitive to insertion order of data points
  - Since we fix the size of leaf nodes, so clusters may not be so natural
  - Clusters tend to be spherical given the radius and diameter measures

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- CHAMELEON: G. Karypis, E. H. Han, and V. Kumar, 1999
- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- Graph-based, and a two-phase algorithm
  1. Use a graph-partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



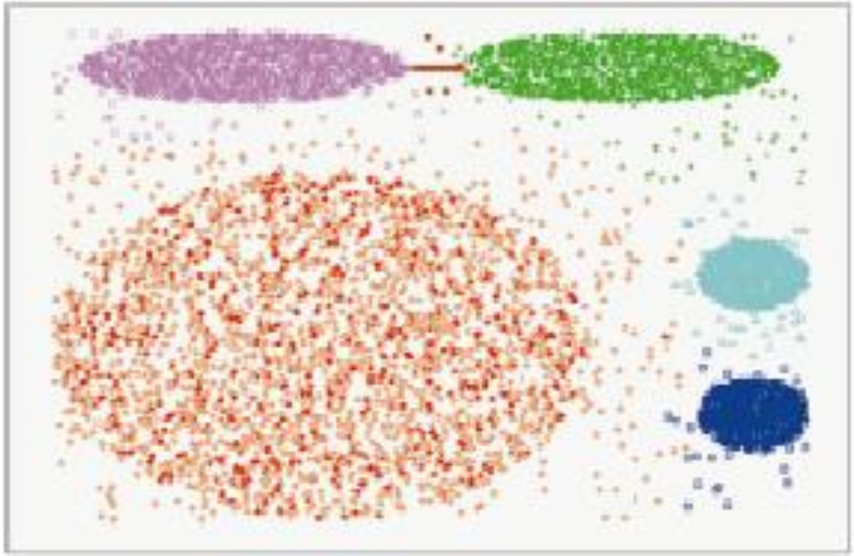
K-NN Graph

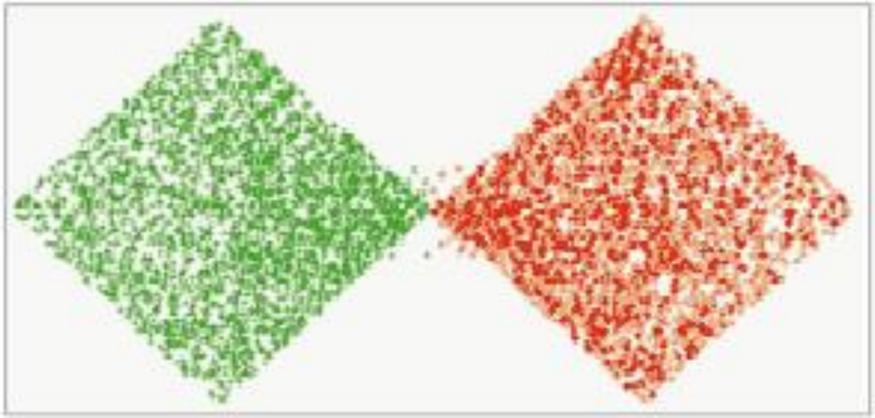
P and q are connected if q is among the top k closest neighbors of p

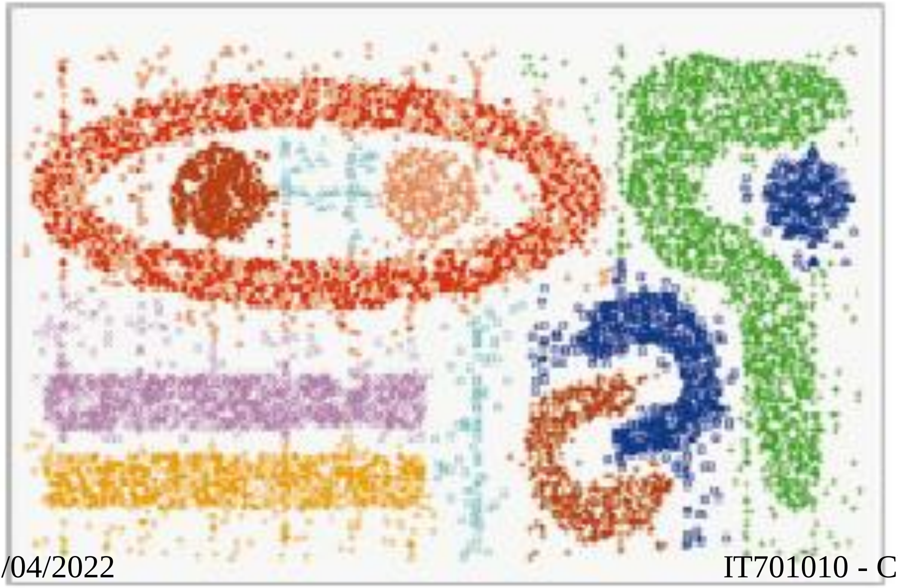
Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

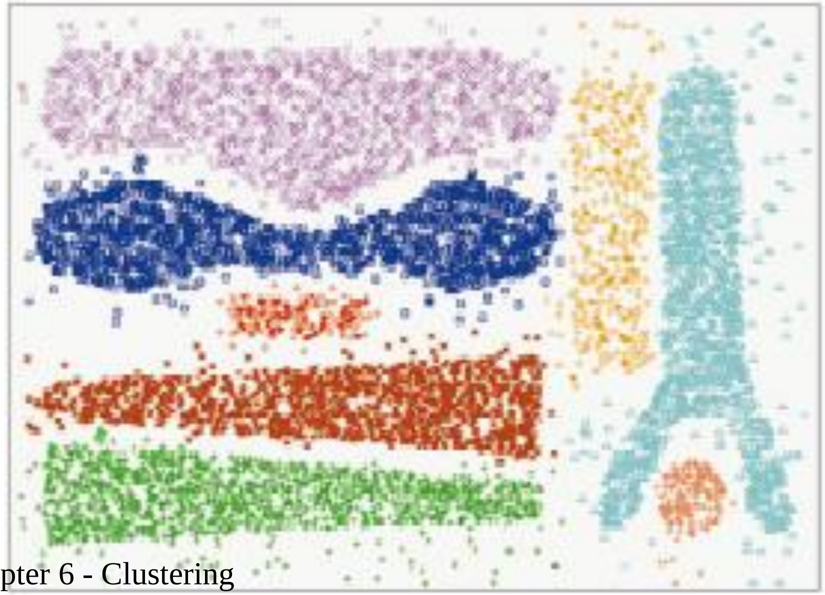
Relative closeness: closeness of c_1 and c_2 over internal closeness

CHAMELEON (Clustering Complex Objects)









Probabilistic Hierarchical Clustering

- Algorithmic hierarchical clustering
  - Nontrivial to choose a good distance measure
  - Hard to handle missing attribute values
  - Optimization goal not clear: heuristic, local search
- Probabilistic hierarchical clustering
  - Use probabilistic models to measure distances between clusters
  - Generative model: Regard the set of data objects to be clustered as a sample of the underlying data generation mechanism to be analyzed
  - Easy to understand, same efficiency as algorithmic agglomerative clustering method, can handle partially observed data
- In practice, assume the generative models adopt common distributions functions, e.g., Gaussian distribution or Bernoulli distribution, governed by parameters

Generative Model

- Given a set of 1-D points $X = \{x_1, \ldots, x_n\}$ for clustering analysis & assuming they are generated by a Gaussian distribution:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability that a point $x_i \in X$ is generated by

- The likelihood that X is generated by the mod

$$P(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The task of learning the generative model: find the parameters μ and σ^2 s

$$L(\mathcal{N}(\mu, \sigma^2) : X) = P(X|\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

the maximum likelihood

$$\mathcal{N}(\mu_0, \sigma_0^2) = \arg \max\{L(\mathcal{N}(\mu, \sigma^2) : X)\}$$

A Probabilistic Hierarchical Clustering Algorithm

- For a set of objects partitioned into *m* clusters $C_1, \dots, C_m$, the quality can be measured by,

$$Q(\{C_1, \dots, C_m\}) = \prod_{i=1}^{m} P(C_i)$$

  where *P*() is the maximum likelihood

- Distance between clusters C_1 and C_2:

$$dist(C_i, C_j) = -\log \frac{P(C_1 \cup C_2)}{P(C_1)P(C_2)}$$

- Algorithm: Progressively merge points and

```
Input: $D = \{o_1, ..., o_n\}$: a data set containing n objects
Output: A hierarchy of clusters
Method
    Create a cluster for each object $C_i = \{o_i\}$, $1 \le i \le n$;
    For i = 1 to n {
        Find pair of clusters $C_i$ and $C_j$ such that
            $C_i, C_j = \text{argmax}_{i \neq j} \{\log (P(C_i \cup C_j)/(P(C_i)P(C_j)))\}$;
        If $\log (P(C_i \cup C_j)/(P(C_i)P(C_j))) > 0$ then merge $C_i$ and $C_j$ }
```

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

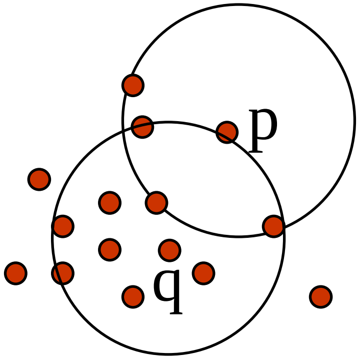
Density-Based Clustering Methods

- 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
- $N_{Eps}(p)$: {q 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:

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

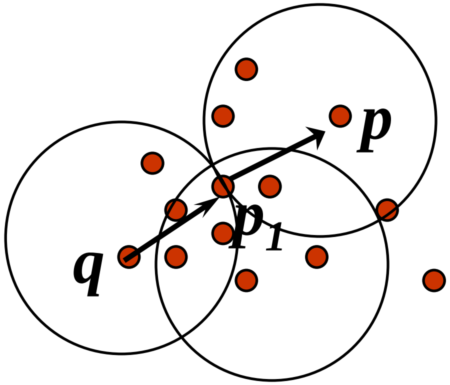


MinPts = 5

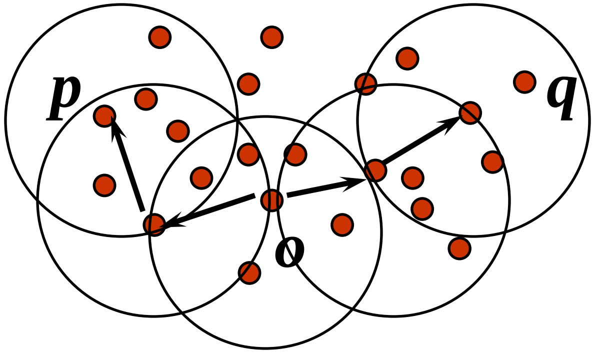
Eps = 1 cm

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

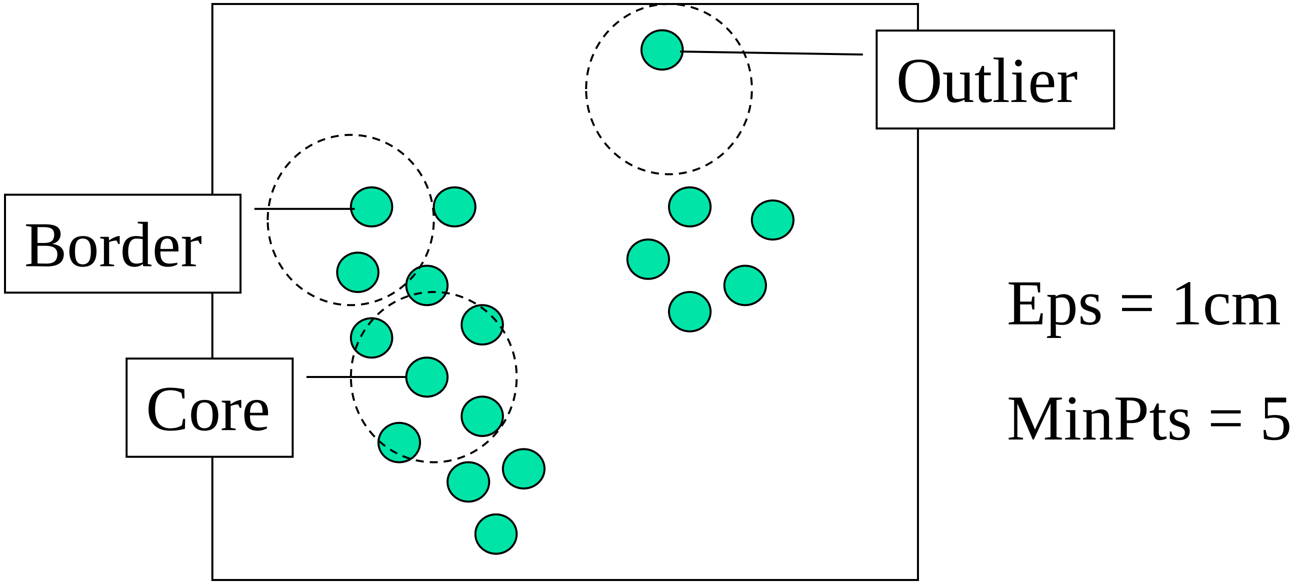


- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



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



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: Sensitive to Parameters

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

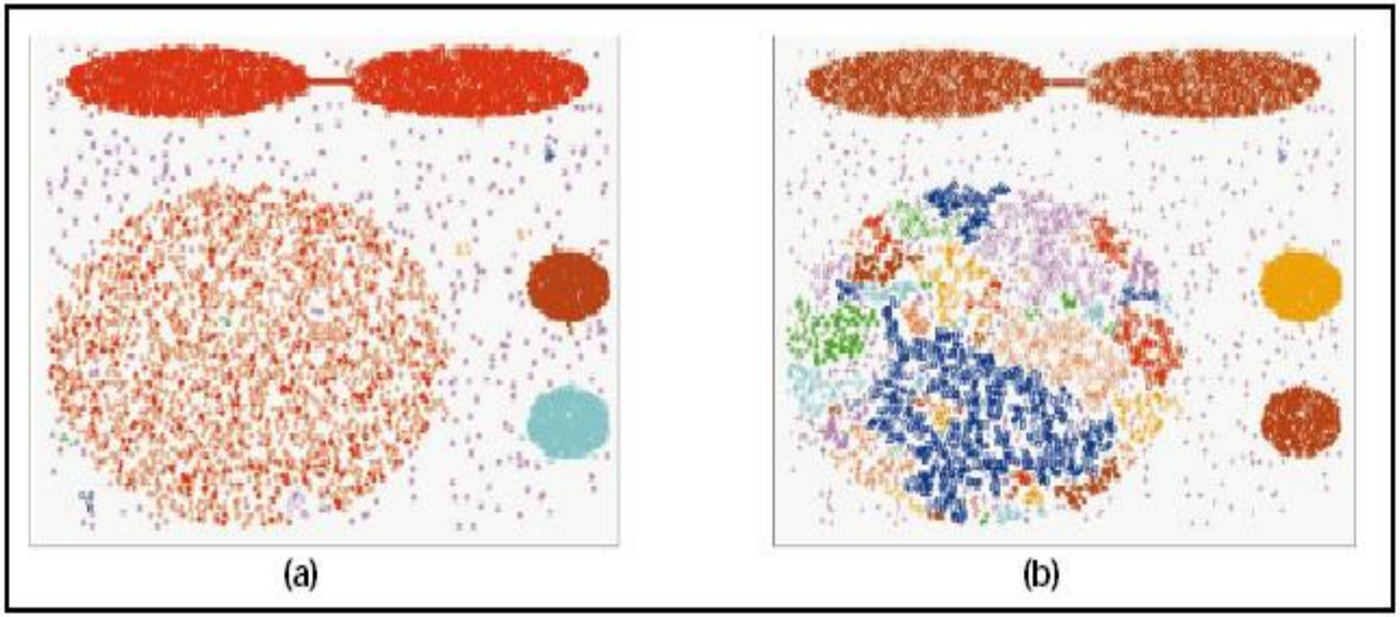
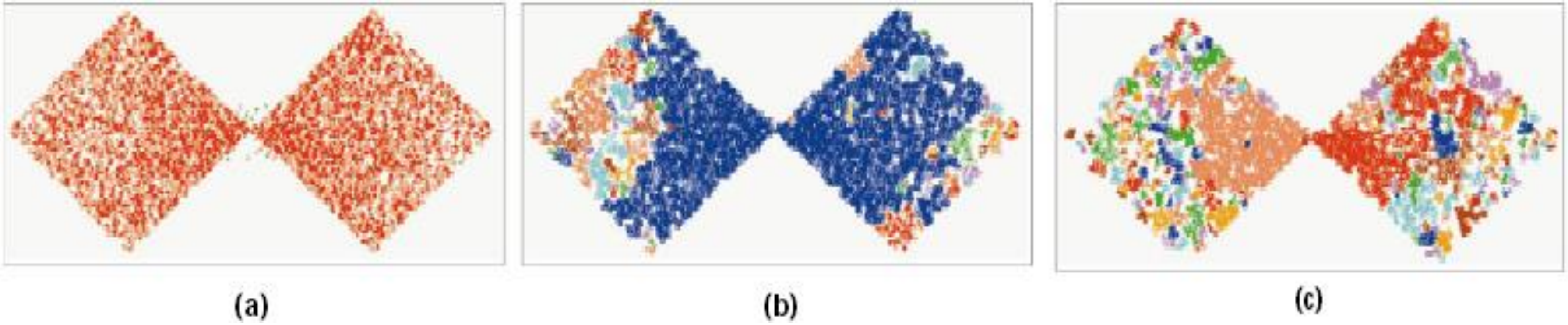


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

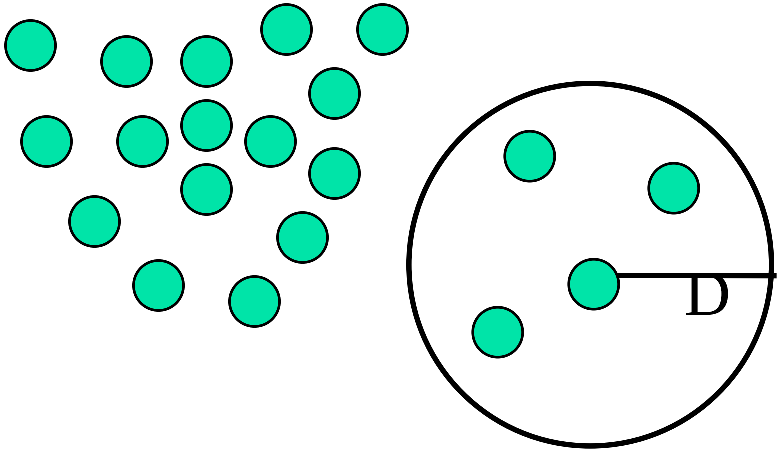


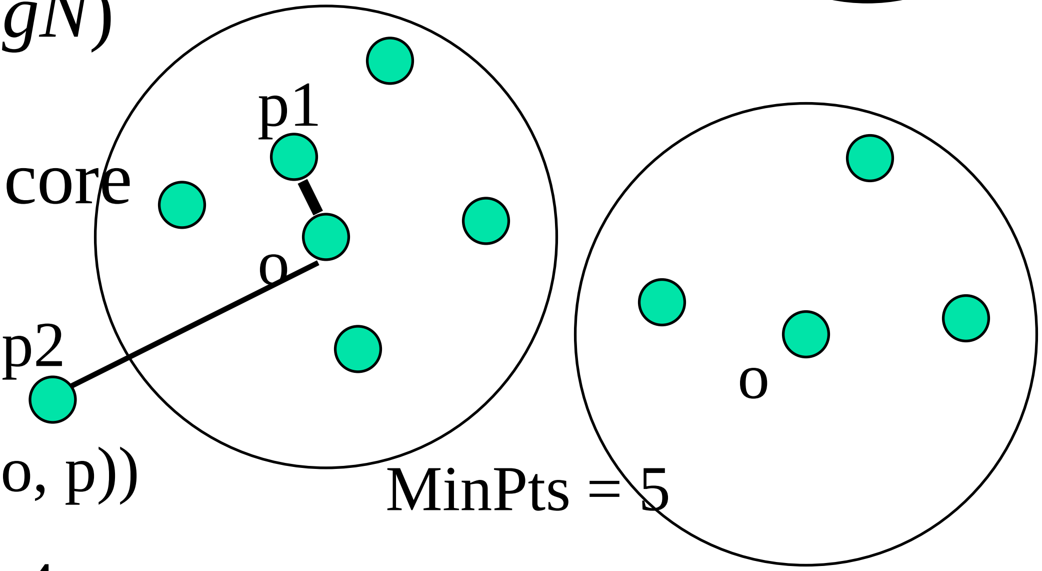
OPTICS: A Cluster-Ordering Method (1999)

- OPTICS: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
  - Produces a special order of the database wrt its density-based clustering structure
  - This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically or using visualization techniques

OPTICS: Some Extension from DBSCAN

- Index-based:
    - k = number of dimensions
    - N = 20
    - p = 75%
    - M = N(1-p) = 5
  - Complexity: O(*NlogN*)
- Core Distance:
  - min eps s.t. point is core
- Reachability Distance



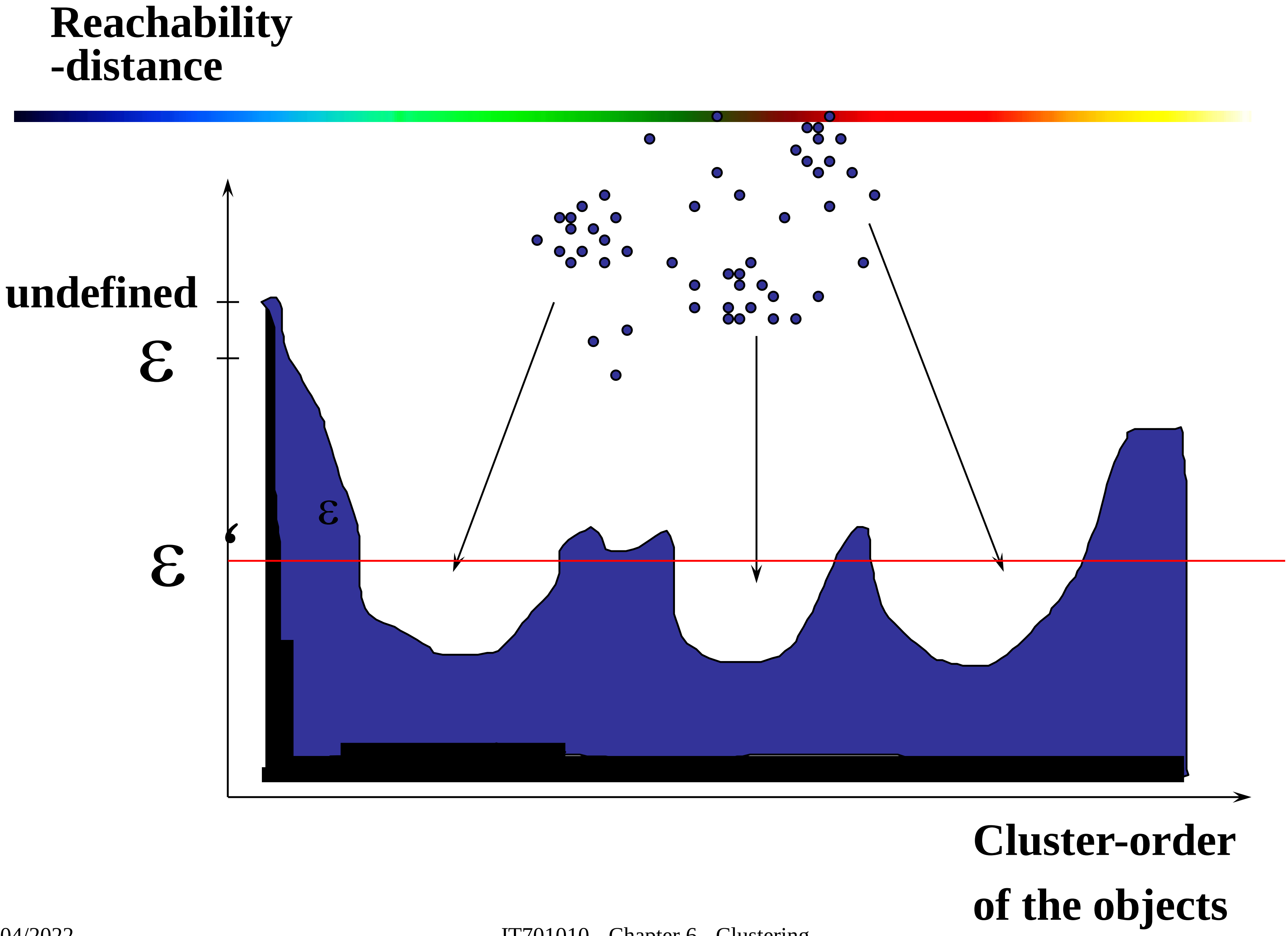


Max (core-distance (o), d (o, p))

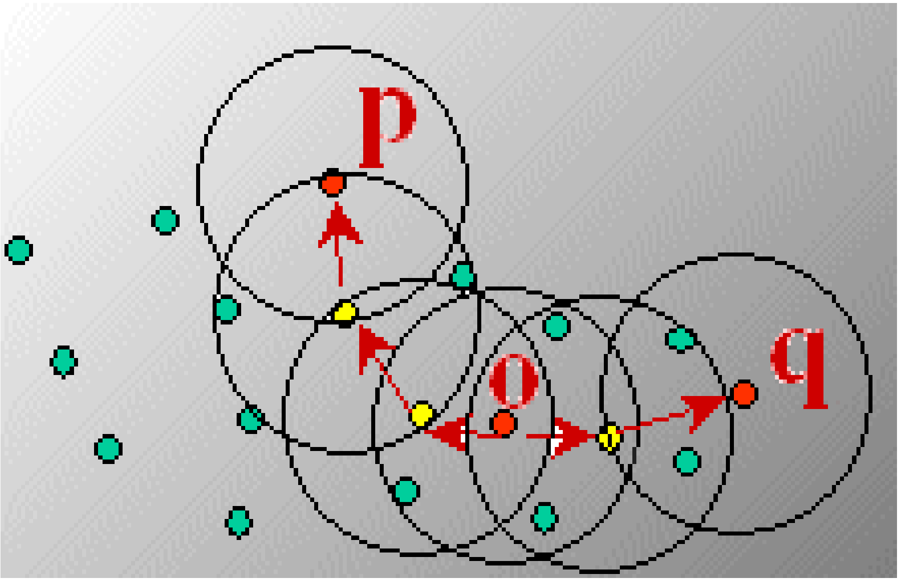
r(p1, o) = 2.8cm. r(p2,o) = 4cm

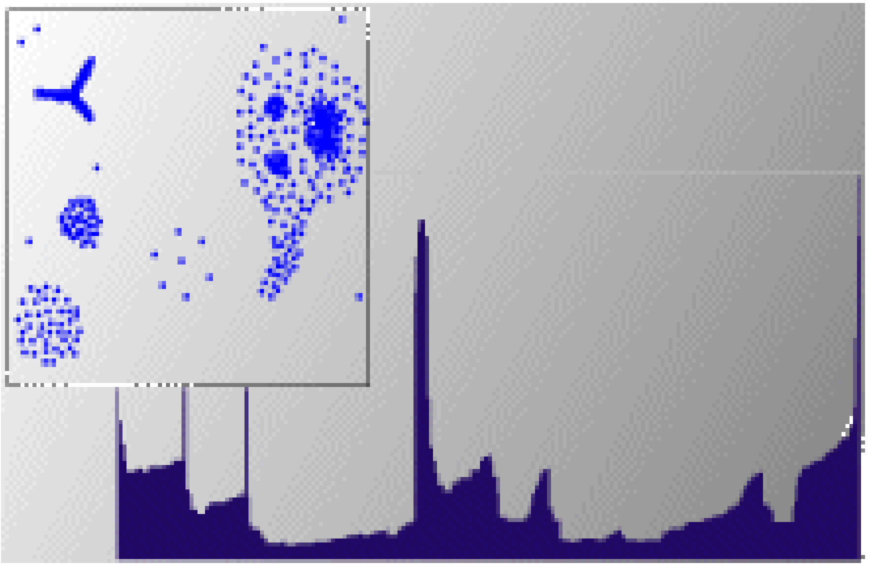
MinPts = 5

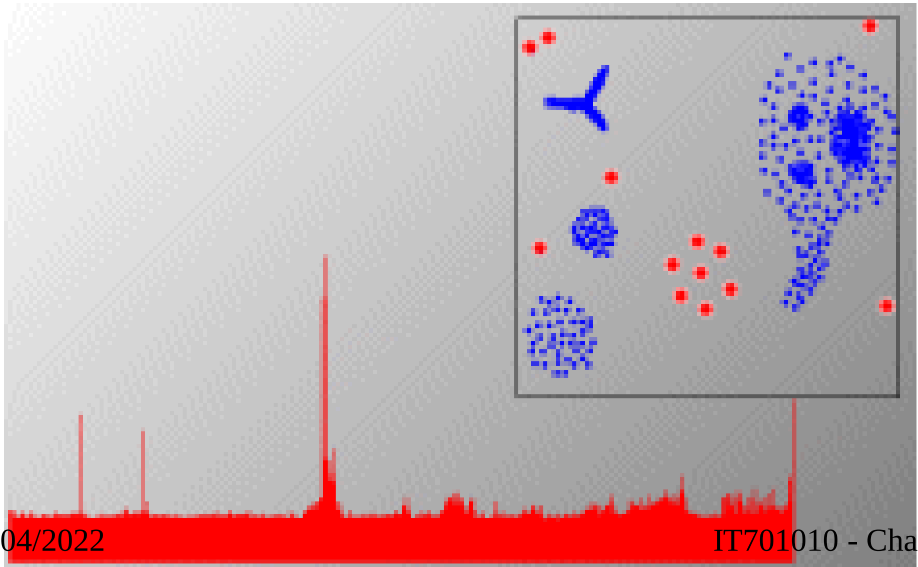
ε = 3 cm

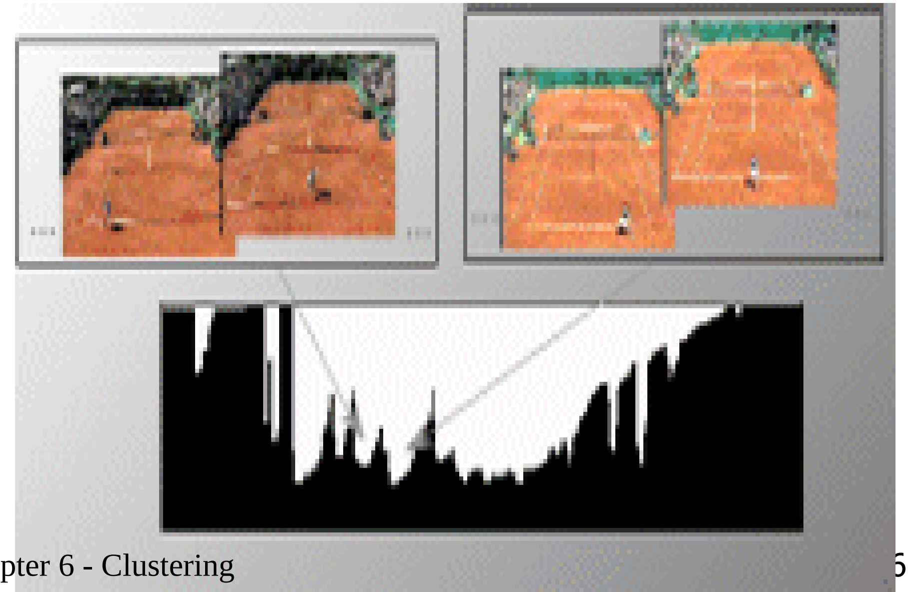

Reachability
-distance
undefined
ε
ε
ε
'
Cluster-order
of the objects

Density-Based Clustering: OPTICS & Its Applications









DENCLUE: Using Statistical Density Functions

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
- Using statistical density functions:

$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

influence of y on x

$$f^{D}_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

total influence on x

$$\nabla f^{D}_{Gaussian}(x, x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

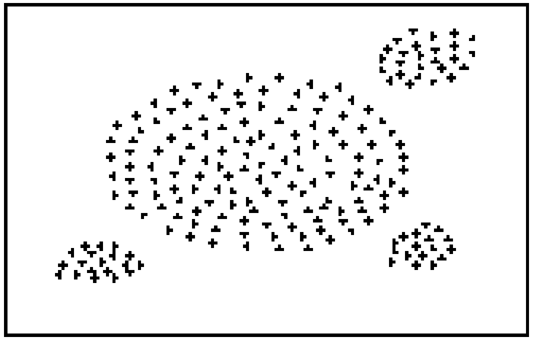
gradient of x in the direction of x_i

- Major features
  - Solid mathematical foundation
  - Good for data sets with large amounts of noise
  - Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
  - Significant faster than existing algorithm (e.g., DBSCAN)
  - But needs a large number of parameters

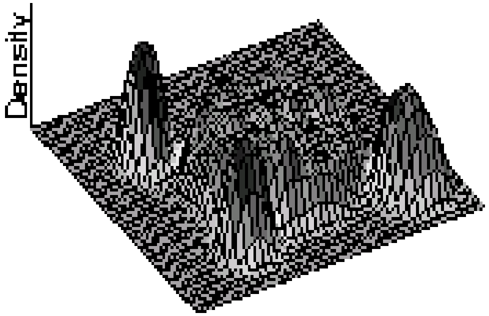
Denclue: Technical Essence

- Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure
- Influence function: describes the impact of a data point within its neighborhood
- Overall density of the data space can be calculated as the sum of the influence function of all data points
- Clusters can be determined mathematically by identifying density attractors
- Density attractors are local maximal of the overall density function
- Center defined clusters: assign to each density attractor the points density attracted to it
- Arbitrary shaped cluster: merge density attractors that are connected through paths of high density (> threshold)

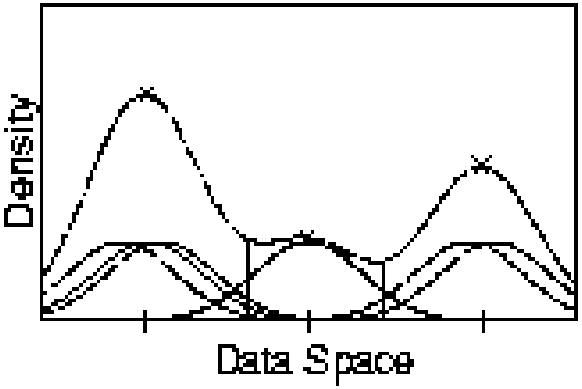
Density Attractor



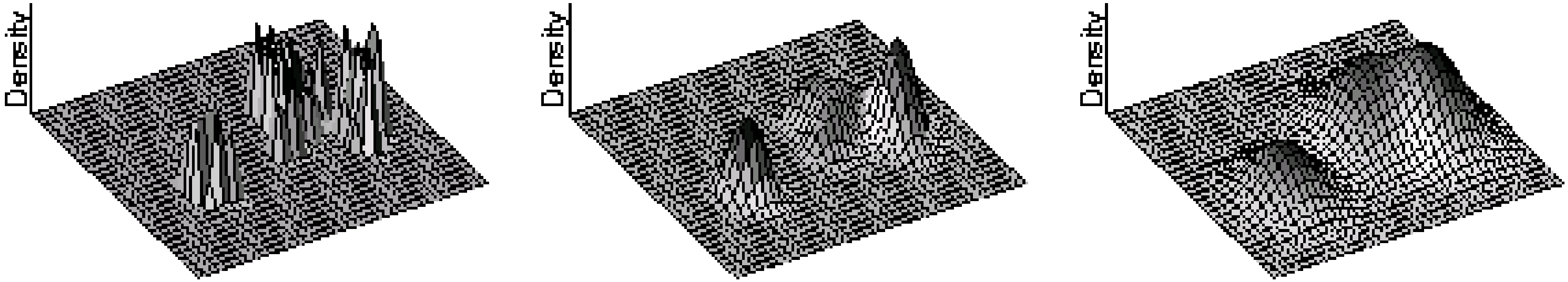
(a) Data Set



(c) Gaussian

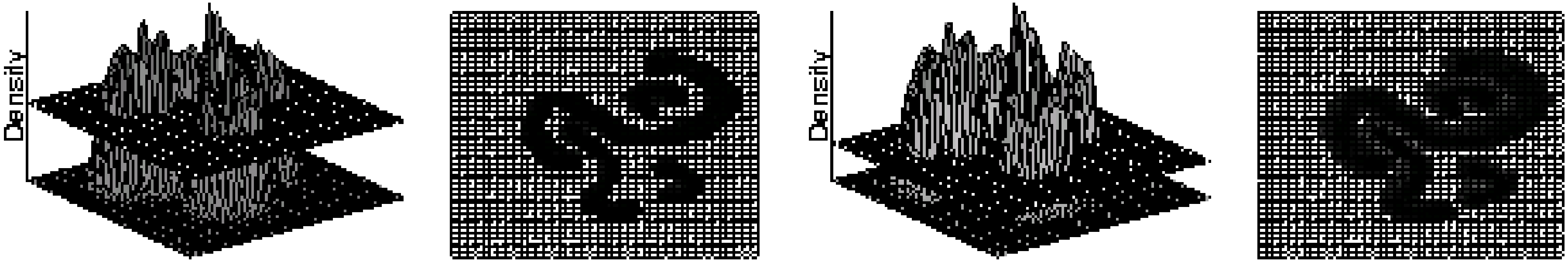


Center-Defined and Arbitrary



(a) $\sigma = 0.2$ (b) $\sigma = 0.6$ (d) $\sigma = 1.5$

Figure 3: Example of Center-Defined Clusters for different σ



(a) $\xi = 2$ (b) $\xi = 2$ (c) $\xi = 1$ (d) $\xi = 1$

Figure 4: Example of Arbitray-Shape Clusters for different ξ

Chapter 10. Cluster Analysis: Basic Concepts and Methods

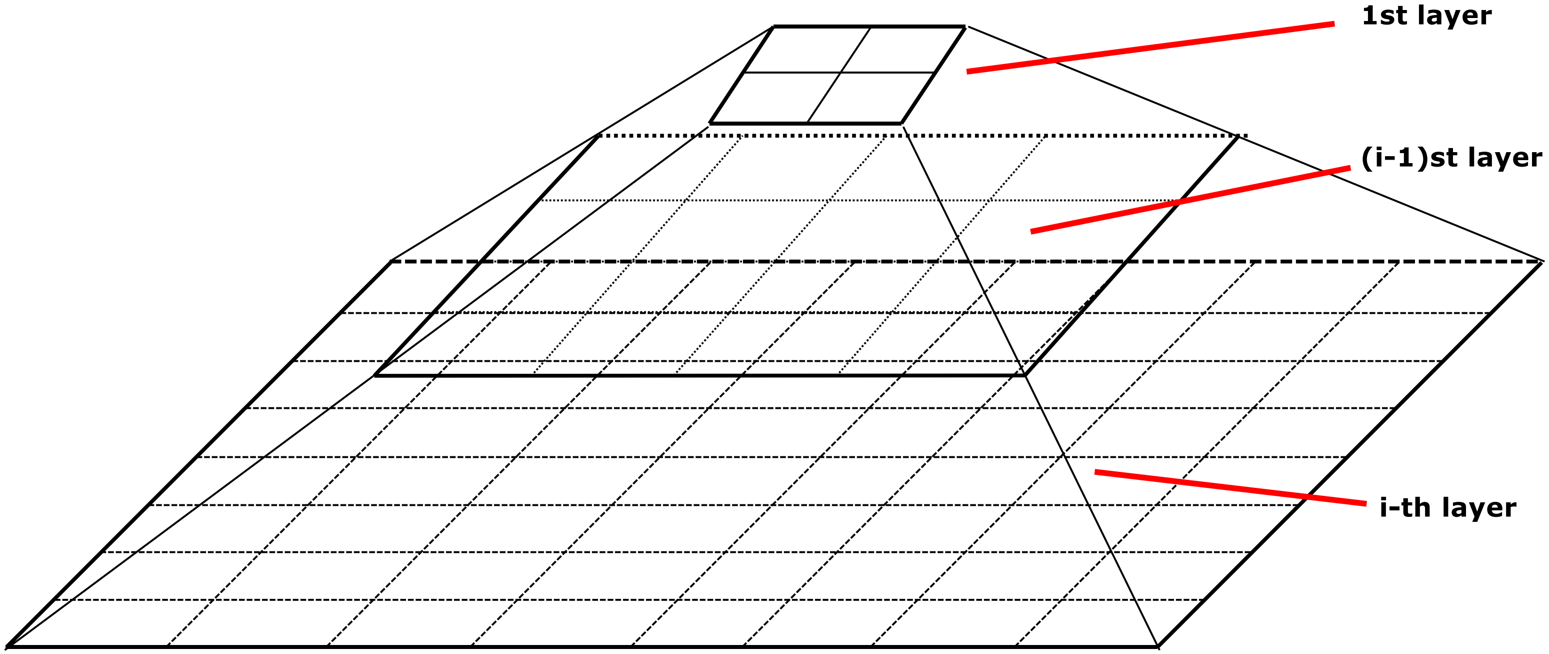
- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

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

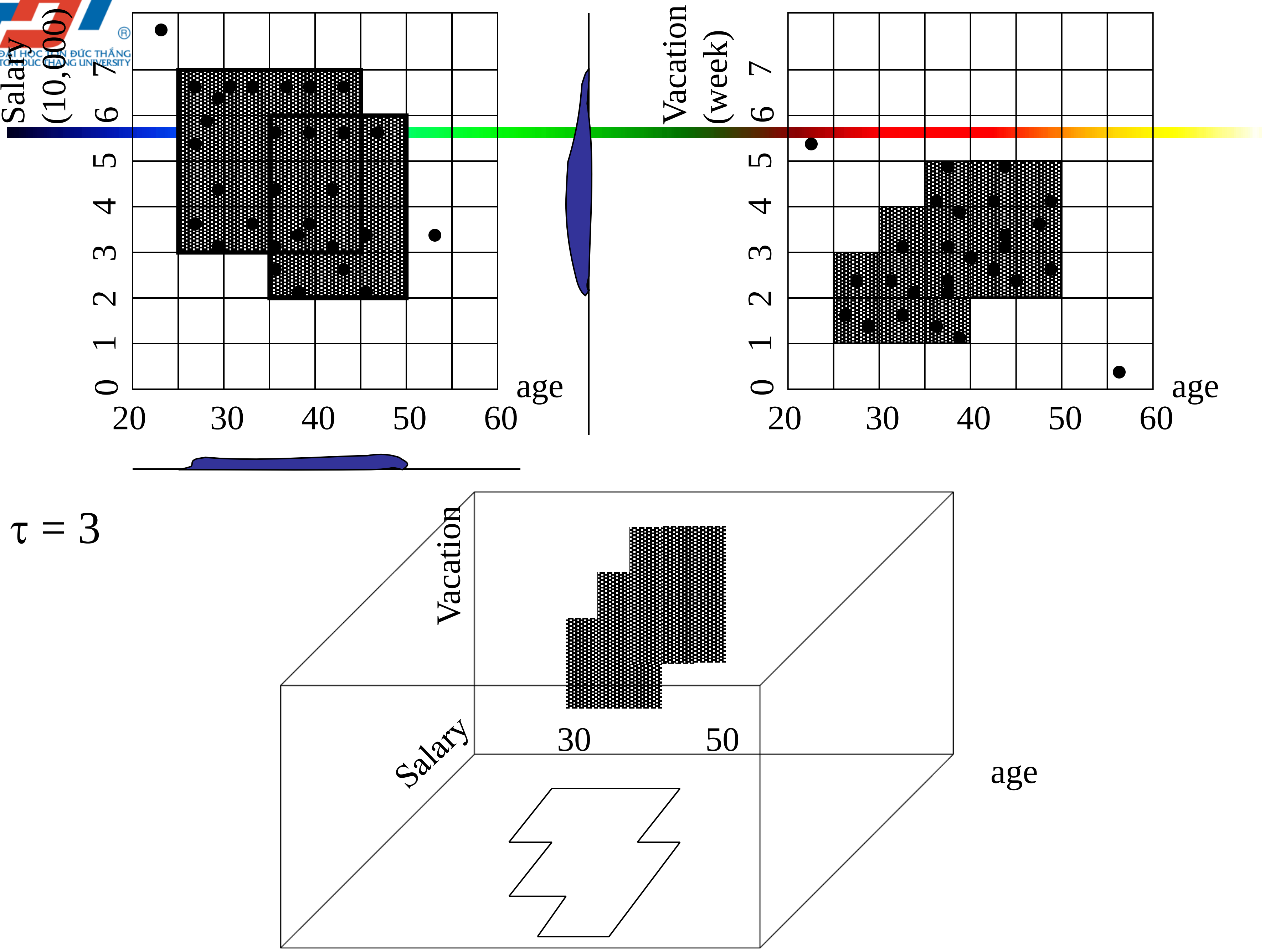
- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O*(*K*), where *K* is the number 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
Vacation
Salary
30
50
age

$\tau = 3$

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

Assessing Clustering Tendency

- Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Test spatial randomness by statistic test: Hopkins Static
  - Given a dataset D regarded as a sample of a random variable o, determine how far away o is from being uniformly distributed in the data space
  - Sample *n* points, $p_1, \ldots, p_n$, uniformly from D. For each p_i, find its nearest neighbor in D: $x_i = min\{dist\,(p_i, v)\}$ where *v* in D
  - Sample *n* points, $q_1, \ldots, q_n$, uniformly from D. For each q_i, find its nearest neighbor in D – $\{q_i\}$: $y_i = min\{dist\,(q_i, v)\}$ where *v* in D and $v \neq q_i$
  - Calculate the Hopkins Statistic:
  - If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5. If D is highly ske

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

Determine the Number of Clusters

- Empirical method
  - # of clusters ≈$\sqrt{n/2}$ 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

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