Data Mining:

Concepts and Techniques

— Chapter 7 —

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering: Rich Applications and Multidisciplinary Efforts

- Spatial Data Analysis
  - Detect spatial clusters or for other spatial mining tasks
- Economic Science (especially market research)
  - Identify customers whose behaviors are similar
- WWW
  - Cluster documents
  - Cluster Weblog data to discover groups of similar access patterns
- Image Processing & Pattern Recognition

Examples of Clustering Applications

- Marketing:
  - Help marketers discover distinct groups in their customer bases
  - Use this knowledge to develop targeted marketing programs
- Land use:
  - Identification of areas of similar land use in an earth observation database
- Insurance:
  - Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning:
  - Identifying groups of houses according to their house type, value, and geographical location

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by 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: $d(i, j)$
- There is a separate "quality" function that measures the "goodness" of a cluster
- The definitions of distance functions
  - Usually very different for interval-scaled, Boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Hard to define "similar enough" or "good enough"
  - The answer is typically highly subjective

Requirements of Clustering in Data Mining

- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with an arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noises and outliers
- Insensitive to the order of input records
- High dimensionality
- Scalability
- Incorporation of user-specified constraints

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

Major Clustering Approaches

- 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, ROCK, CHAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS

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 an 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 squared distances between all pairs of points in the cluster

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

Typical Alternatives to Calculate the Distance between Clusters

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

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. Summary

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, having min sum of squared distances of objects to their representative of a cluster

$$\Sigma_{m=1}^{k}\Sigma_{t_{mi}\in Km}(C_m - t_{mi})^2$$

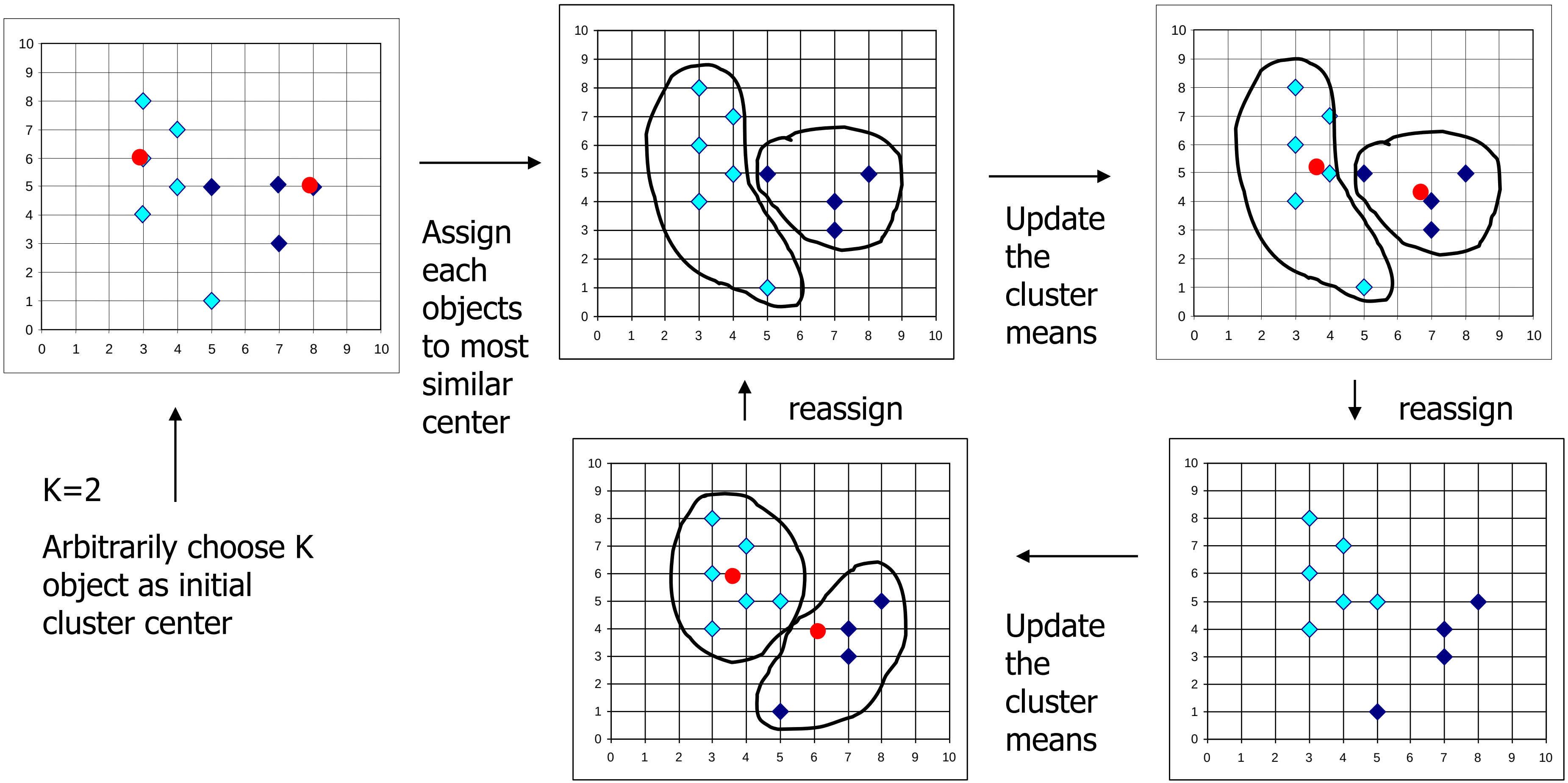
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *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 partition (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 no more new assignment

The *K-Means* Clustering Method

- Example



Comments on the *K-Means* Method

- Strength: *Relatively 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 optimum*. The *global optimum* may be found using other techniques such as *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined (what about categorical data?)
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noises and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- Handling categorical data: *k-modes* (Huang'98)
  - Idea: replacing means of clusters with modes
    - X, Y: objects having m categorical attributes
    - Dissimilarity d(X,Y): the number of total mismatches

$$d(X,Y)=\Sigma_{j=1}^{m}\delta(x_j,y_j) \text{ where } \delta(x_j,y_j)=\begin{cases}0 & (x_j = y_j)\\ 1 & (x_j \neq y_j)\end{cases}$$

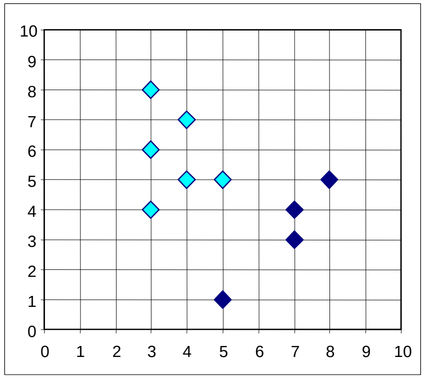
    - *Mode* of X = {X1, X2, …, Xn} is a vector Q = <q1, q2, …., qm> that minimizes

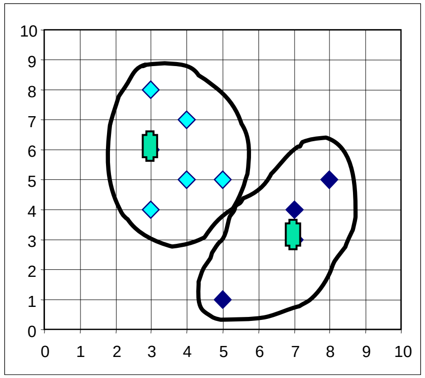
$$D(X,Q)=\Sigma_{i=1}^{n}d(X_i,Q)$$

    - Finding a mode for X
      - Taking the value most frequently occurring for each attribute
      - 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 (i.e., *centroids*) of the object in a cluster as a reference point, *medoids* can be used, which is the *most centrally-located object* in a cluster.





The *K-Medoids* Clustering Method

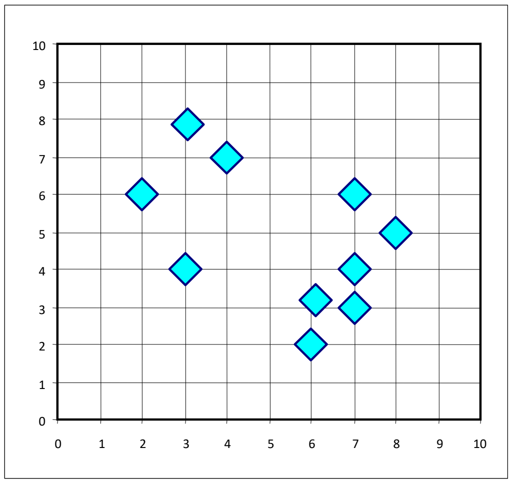
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

PAM (Partitioning Around Medoids) (1987)

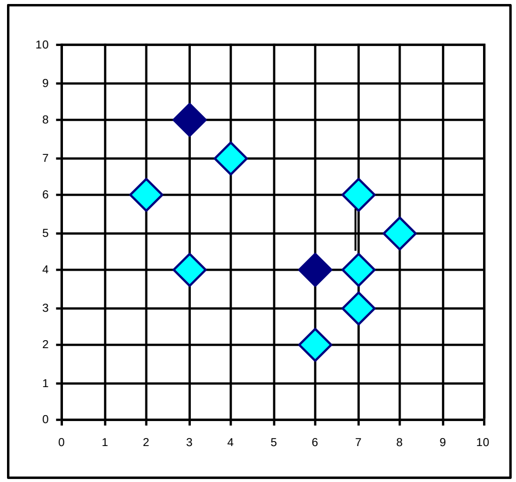
- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use a real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and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

A Typical K-Medoids Algorithm (PAM)

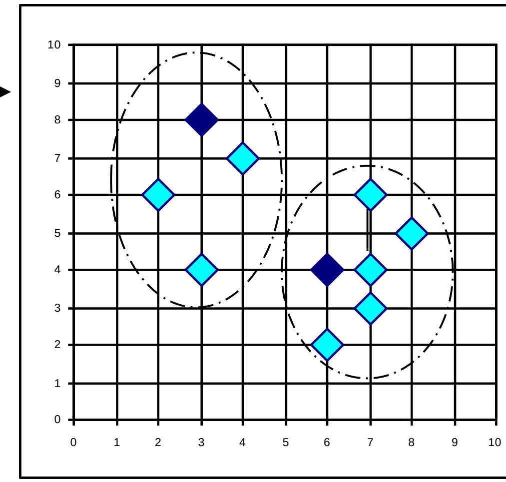
Total Cost = 20



Arbitrary choose k object as initial medoids



Assign each remainin g object to nearest medoids



K=2

Randomly select a nonmedoid object, O_{ramdom}

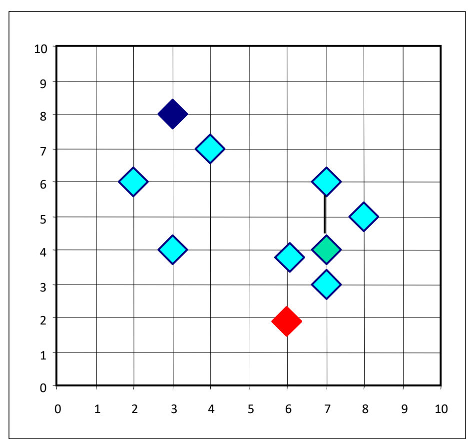
Total Cost = 26

Do loop

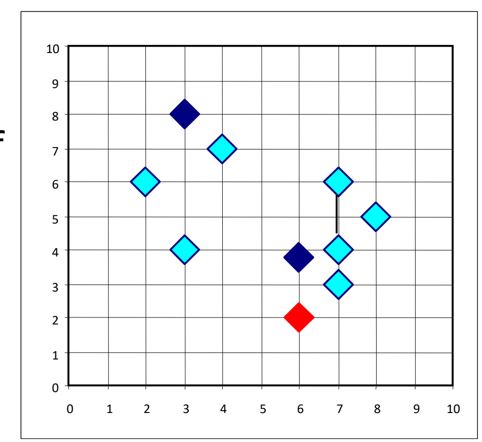
Until no change

Swapping O and O_{ramdom}

If quality is improved.



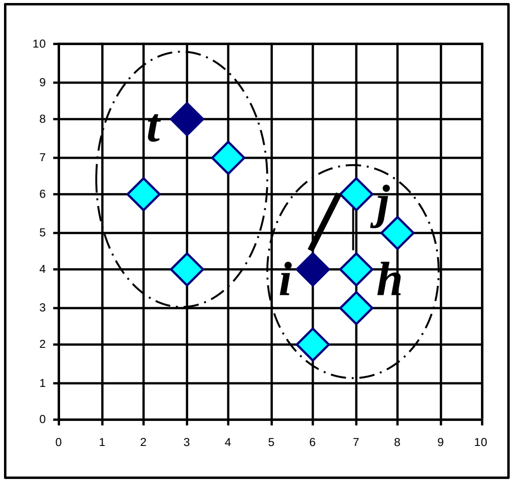
Compute total cost of swapping



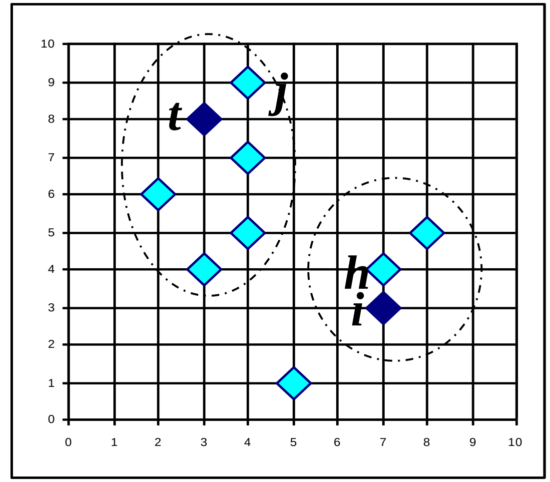
PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

NewC - OldC

- i: original seed
- h: new seed
- t: other seed
- j: non-seed

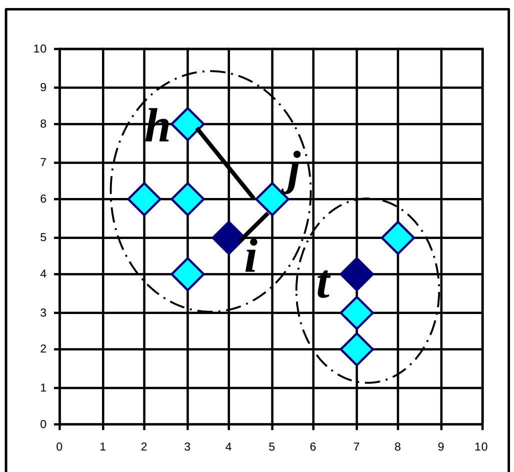


$$C_{jih} = d(j, h) - d(j, i)$$

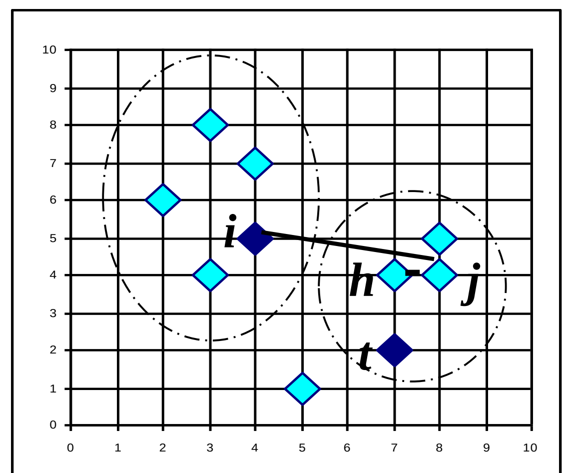


$$C_{jih} = 0$$

to i and now belongs to h
to t and again belongs to t
to i and now belongs to t
to t and now belongs to h



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



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

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.
  - $O(i*k*(n-k)^2)$ where *n* is # of data, *k* is # of clusters, *i* is # of iterations

➔ Sampling based method,

CLARA (Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

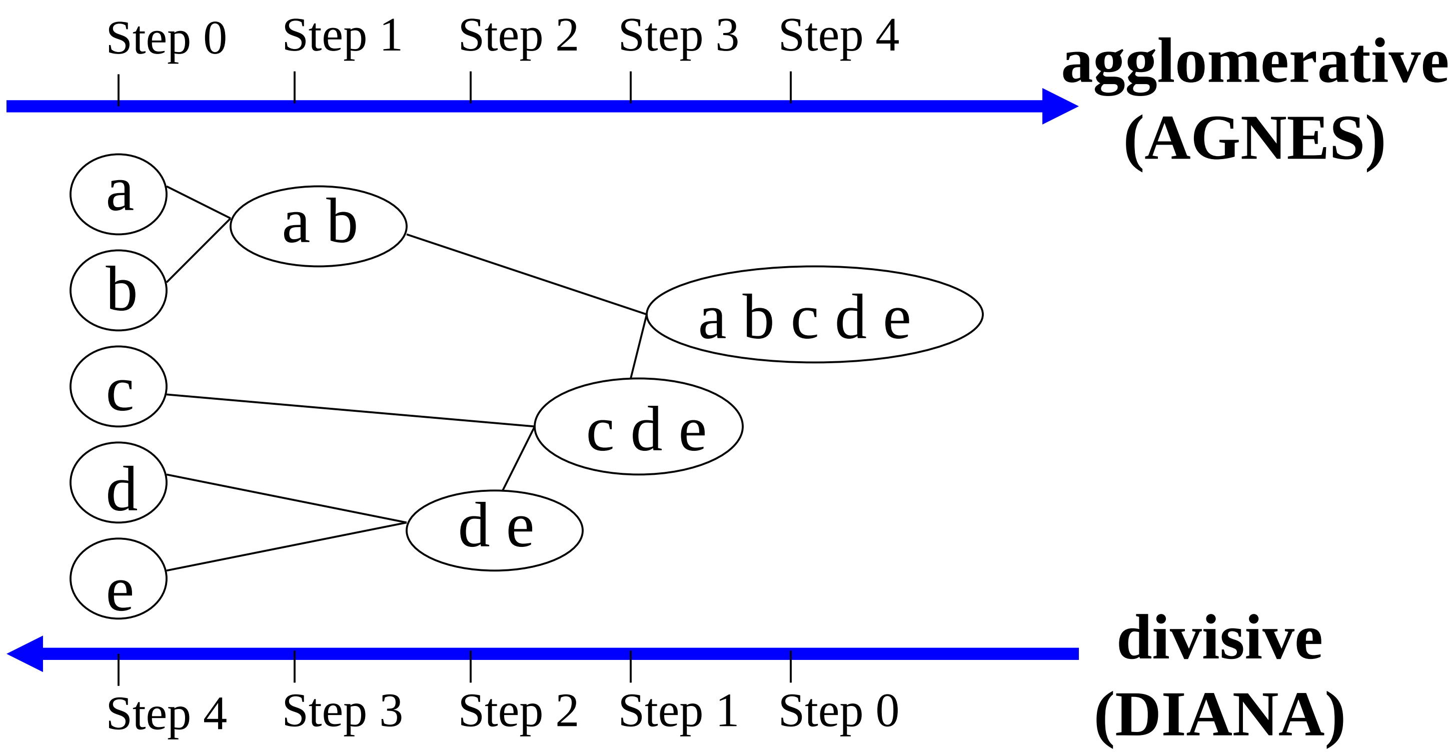
- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency *depends 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*

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. Summary

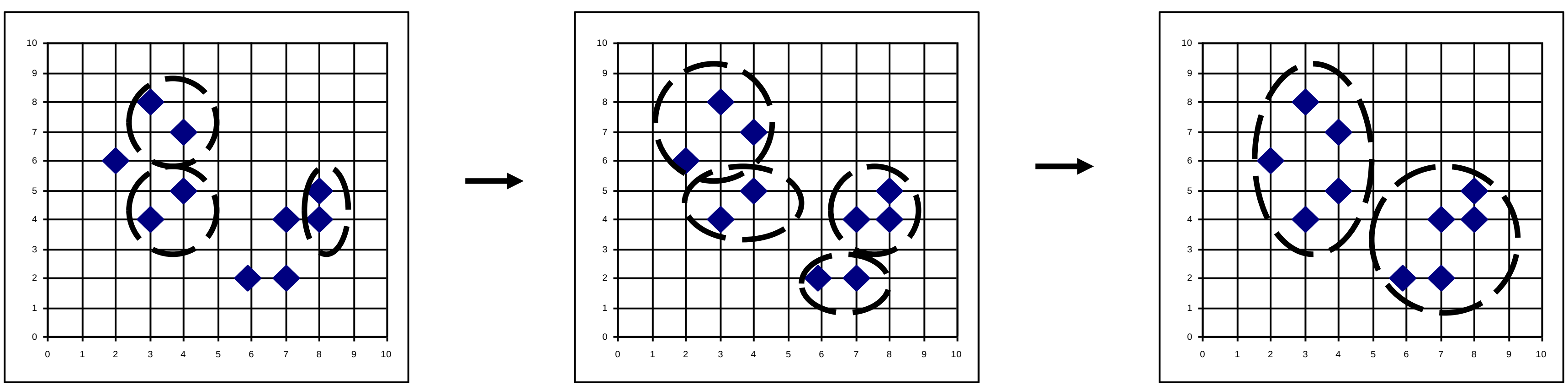
Hierarchical Clustering

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

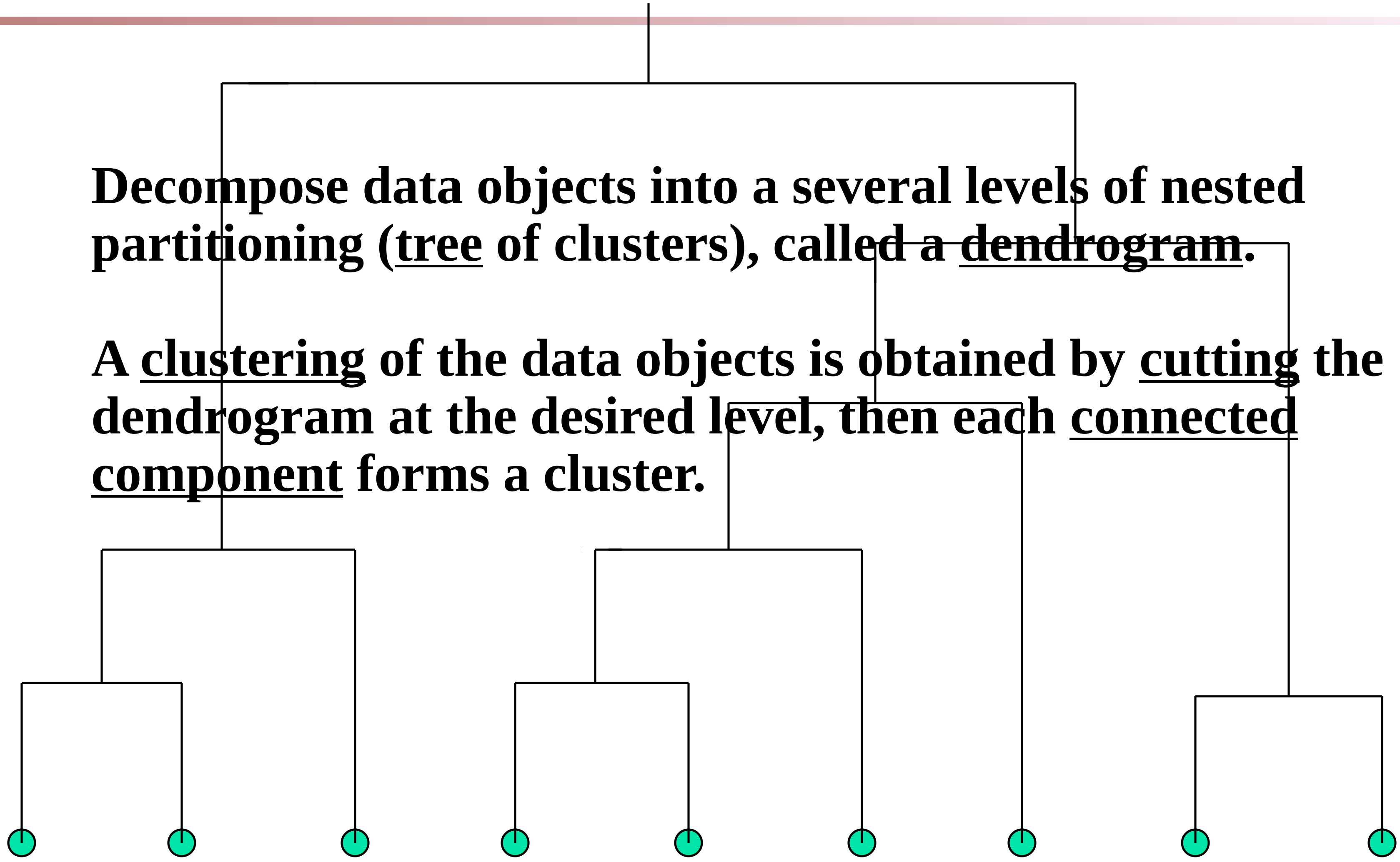


AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
  - Implemented in statistical analysis packages, 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: How the Clusters are Merged

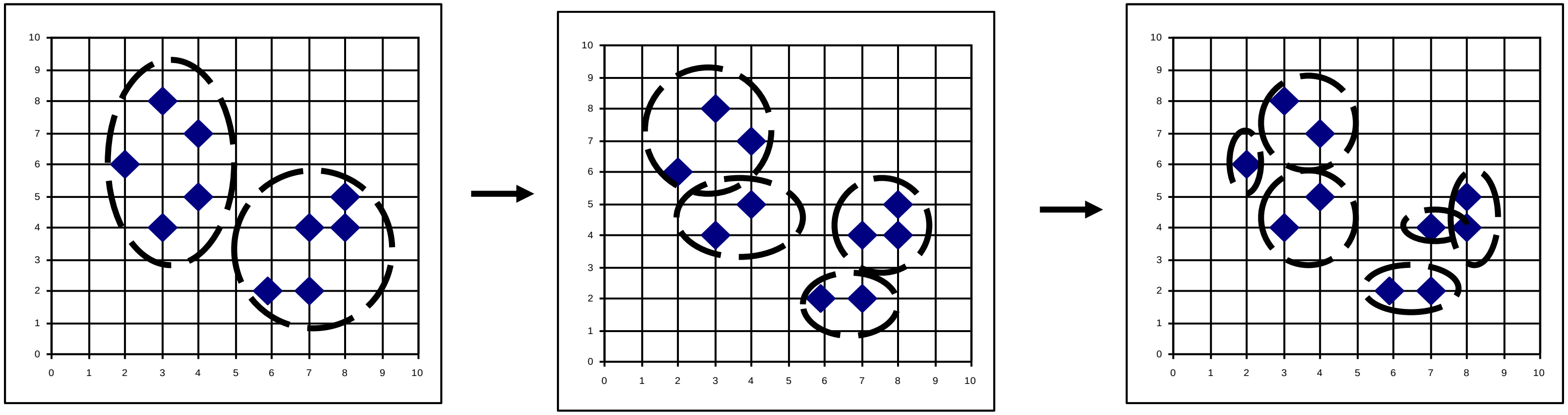


Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.

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

DIANA (Divisive Analysis)

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



DIANA (Divisive Analysis)

- Outline
  - Initially, there is one large cluster consisting of all *n* objects
  - At each subsequent step, the largest available cluster is split into two clusters
    - Until finally all clusters comprise of a single object.
    - Thus, the hierarchy is built in *n*-1 steps.
- Complexity in the first step
  - Agglomerative method: $\frac{n(n-1)}{2}$ possible combinations
  - Divisive method: $2^{n-1}-1$ possible combinations
    - Considerably larger than an agglomerative method

DIANA (Divisive Analysis)

- To avoid considering all possibilities, the algorithm proceeds as follows.
  1. Find the object, which has the highest average dissimilarity to all other objects. This object initiates a new cluster– a sort of a *splinter group*.
  2. For each object *i* outside the *splinter group*, compute
     $D_i = [average\ d(i,j)\ j \notin R_{splinter\ group}] - [average\ d(i,j)\ j \in R_{splinter\ group}]$
  3. Find an object *h* for which the difference D_h is the largest. If D_h is positive, then *h* is, on the average close to the splinter group. Put h into the splinter group.
  4. Repeat *Steps* 2 and 3 until all differences D_h are negative. The data set is then split into two clusters.
  5. Select the cluster with the largest diameter. The diameter of a cluster is the largest dissimilarity between any two of its objects. Then divide this cluster, following steps 1-4.
  6. Repeat *Step* 5 until all clusters contain only a single object.

Advacned Hierarchical Clustering Methods

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

BIRCH (1996)

- 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

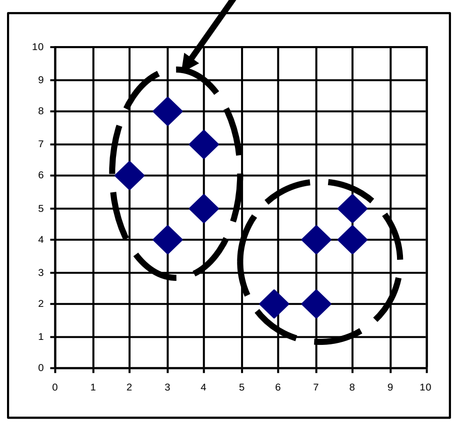
Clustering Feature: ***CF* = (*N*, $\overrightarrow{LS}$, *SS*)**

N: **Number of data points**

LS: $\sum^{N}_{i=1} = \vec{X}_i$

SS: $\sum^{N}_{i=1} = \vec{X}_i^{2}$

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



(3,4)

(2,6)

(4,5)

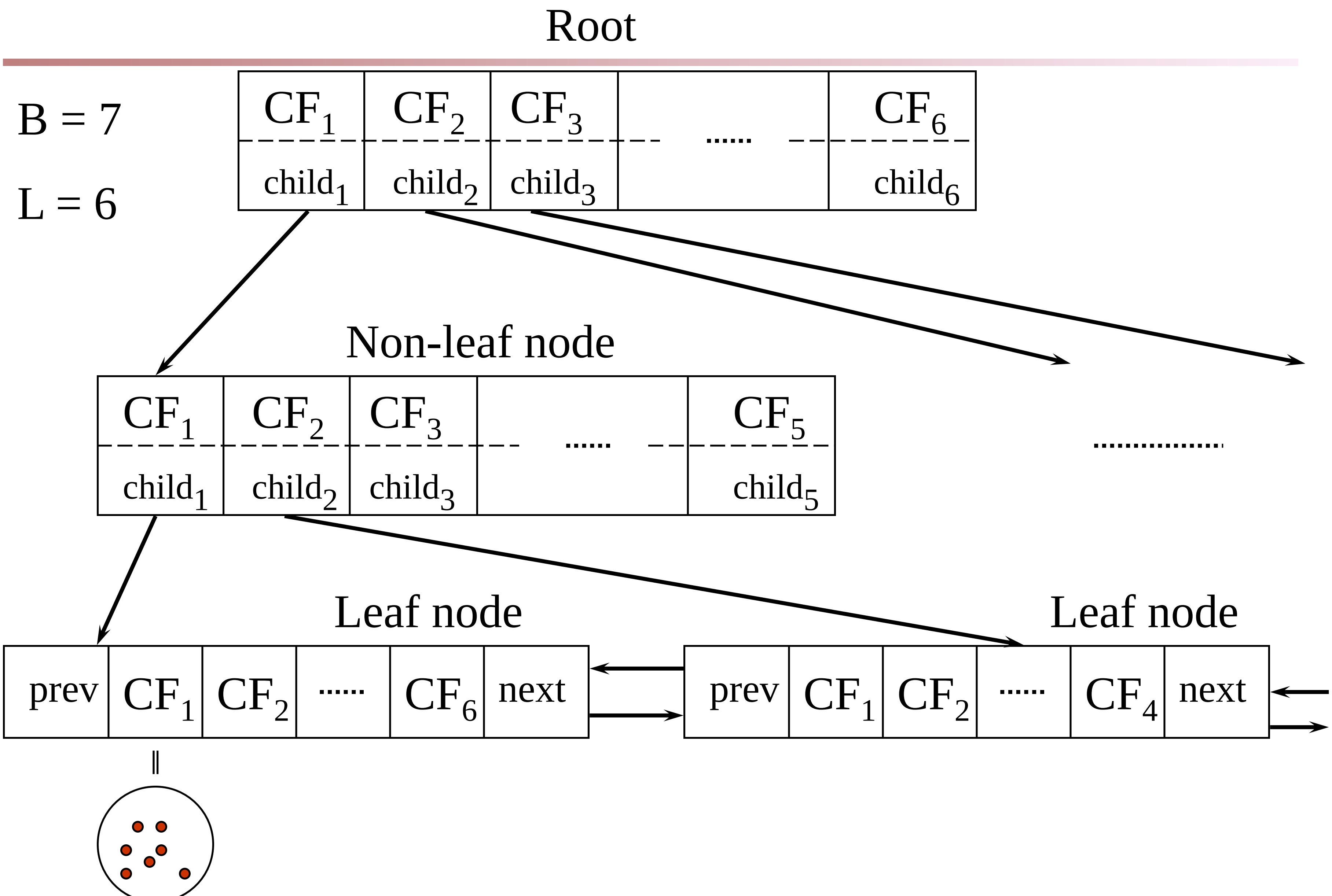
(4,7)

(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 non-leaf node in a tree has descendants or "children"
  - A non-leaf node stores the sum of the CFs of their children
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children
  - threshold: max diameter of a sub-cluster stored at the leaf node

The CF Tree Structure



Clustering Categorical Data: The ROCK Algorithm

- ROCK: RObust Clustering using linKs
  - S. Guha, R. Rastogi & K. Shim, ICDE'99
- Major ideas
  - Use *links* to measure similarity/proximity
    - Not distance-based

Similarity Measure in ROCK

- Traditional measures for categorical data may not work well, e.g., Jaccard coefficient
- Example: Two groups (clusters) of transactions
  - C_1. <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2. <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Jaccard coefficient may lead to a wrong clustering result
  - C_1: 0.2 ({a, **b**, c}, {**b**, d, e}} to 0.5 ({**a, b**, c}, {**a, b**, d})
  - C_1 & C_2: could be as high as 0.5 ({**a, b**, c}, {**a, b**, f})
- *Jaccard coefficient*-based similarity function:

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

  - Ex. Let T_1 = {a, b, c}, T_2 = {c, d, e}

$$Sim(T_1, T_2) = \frac{|\{c\}|}{|\{a, b, c, d, e\}|} = \frac{1}{5} = 0.2$$

Link Measure in ROCK

- Links: # of common *neighbors* (threshold = 0.5 in jC)
  - C_1 <a, b, c, d, e>: **{a, b, c}**, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, **{c, d, e}**
  - C_2 <a, b, f, g>: **{a, b, f}**, {a, b, g}, {a, f, g}, {b, f, g}
- Let T_1 = {**a**, **b**, **c**}, T_2 = {**c**, d, e}, T_3 = {**a**, **b**, f}
  - link(T_1, T_2) = 4, *since they have 4 common neighbors*
    - {a, c, d}, {a, c, e}, {b, c, d}, {b, c, e}
  - link(T_1, T_3) = 3, *since they have 3 common neighbors*
    - {a, b, d}, {a, b, e}, {a, b, g}
- Thus, link is a better measure than Jaccard coefficient

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- CHAMELEON: by G. Karypis, E.H. Han, and V. Kumar'99
- 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 internal closeness of items within the clusters

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- Draw a k-nearest neighbor graph first
  - Node: object, edge: k-nearest neighbor's link, weight: similarity
- A two-phase algorithm
  - Use a graph partitioning algorithm:
    - Cluster objects into a large number of relatively small sub-clusters
  - Use an agglomerative hierarchical clustering algorithm:
    - Find the genuine clusters by repeatedly combining these sub-clusters

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

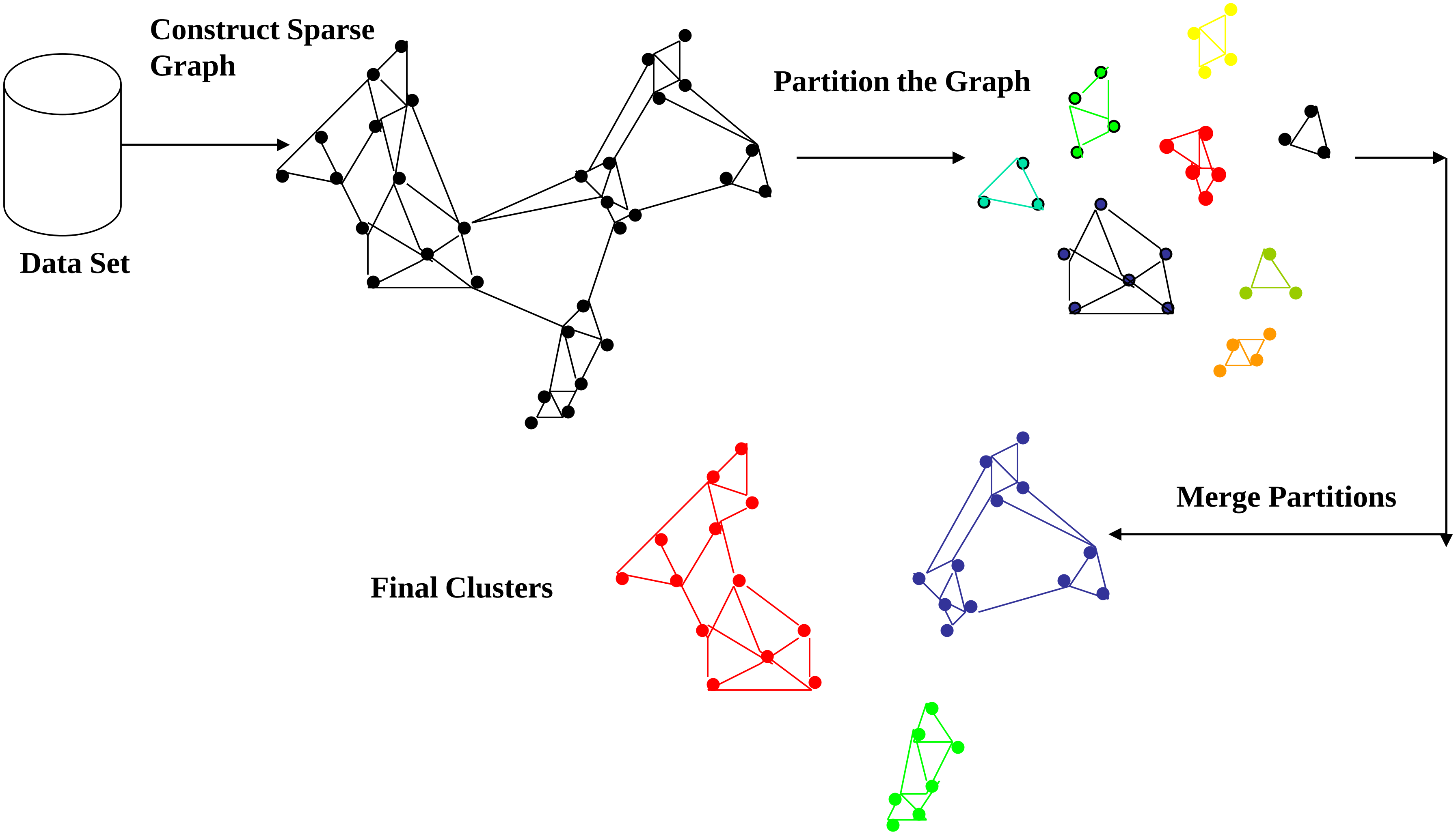
- Partitioning
  - To minimize the edge cut (**METIS**)
    - Tries to split a graph into two subgraphs of nearly equal sizes
- Relative interconnectivity

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)},$$

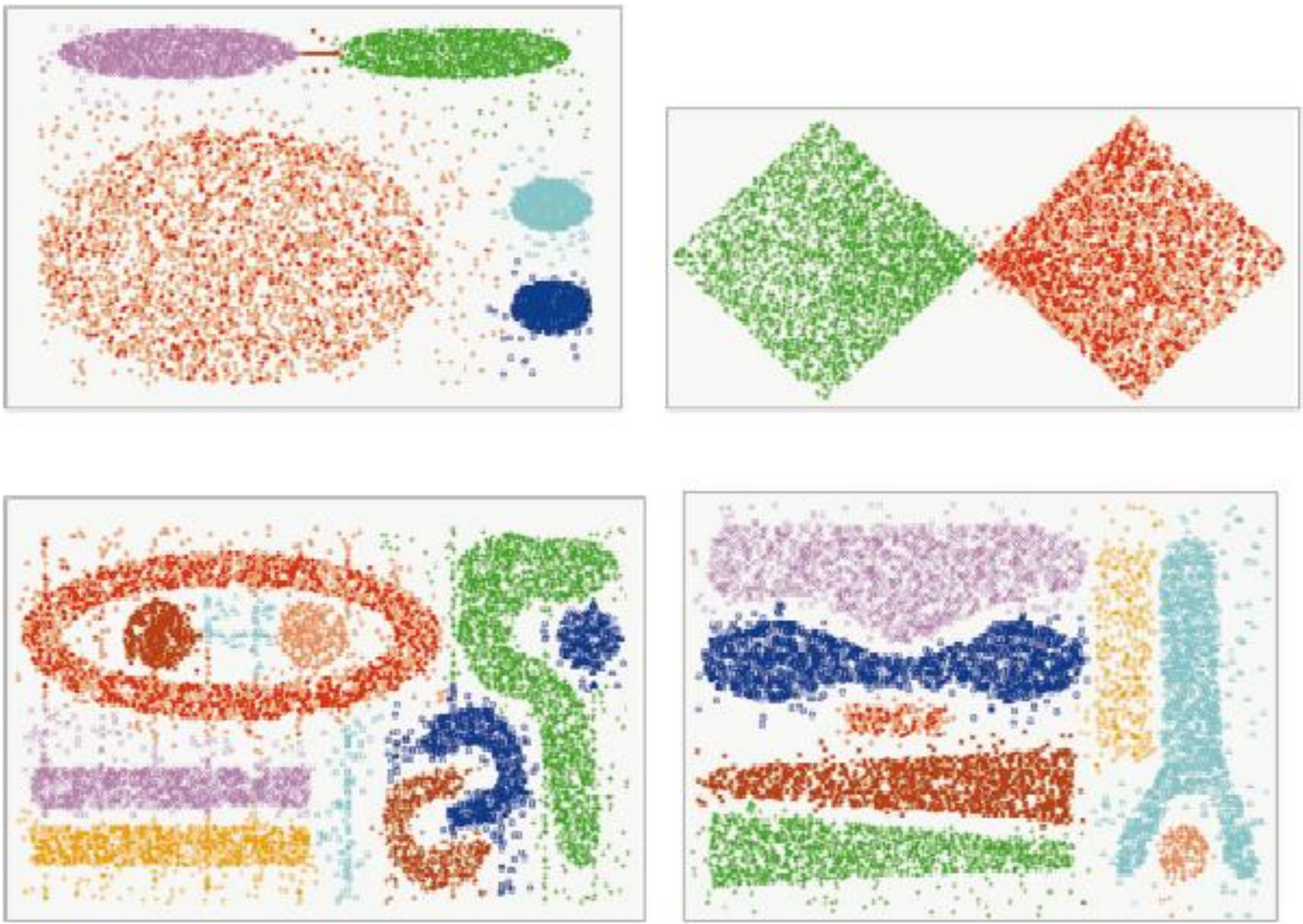
- Relative closeness

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}},$$

Overall Framework of CHAMELEON



CHAMELEON (Clustering Complex Objects)



Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. Summary

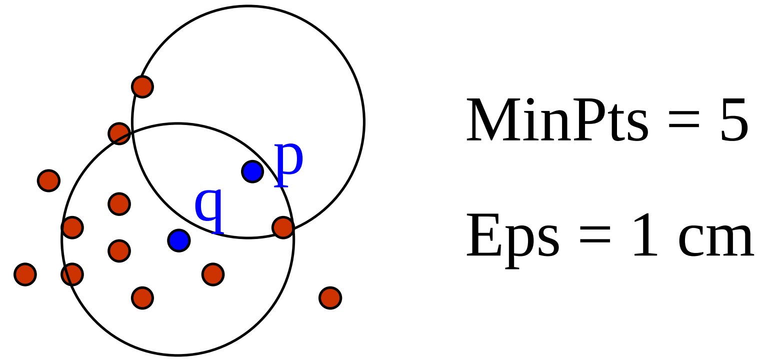
Density-Based Clustering Methods

- 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

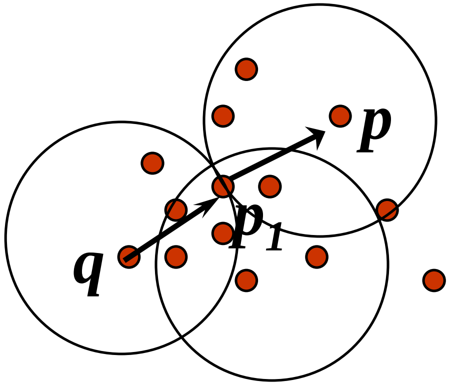
- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an Eps-neighborhood of a given point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t.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

    $|N_{Eps}(q)| >= MinPts$
  - Note: *Not symmetric*

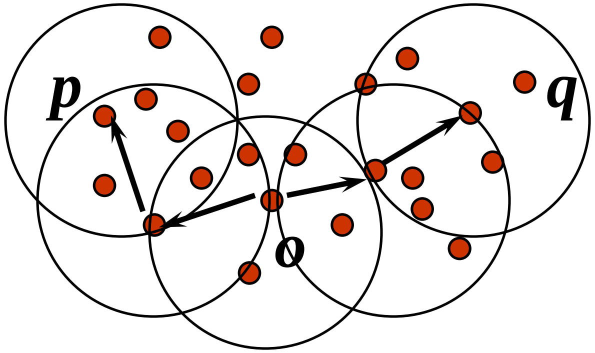


Density-Reachable and Density-Connected

- Density-reachable:
  - A point p is density-reachable from a point q w.r.t. *Eps and 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

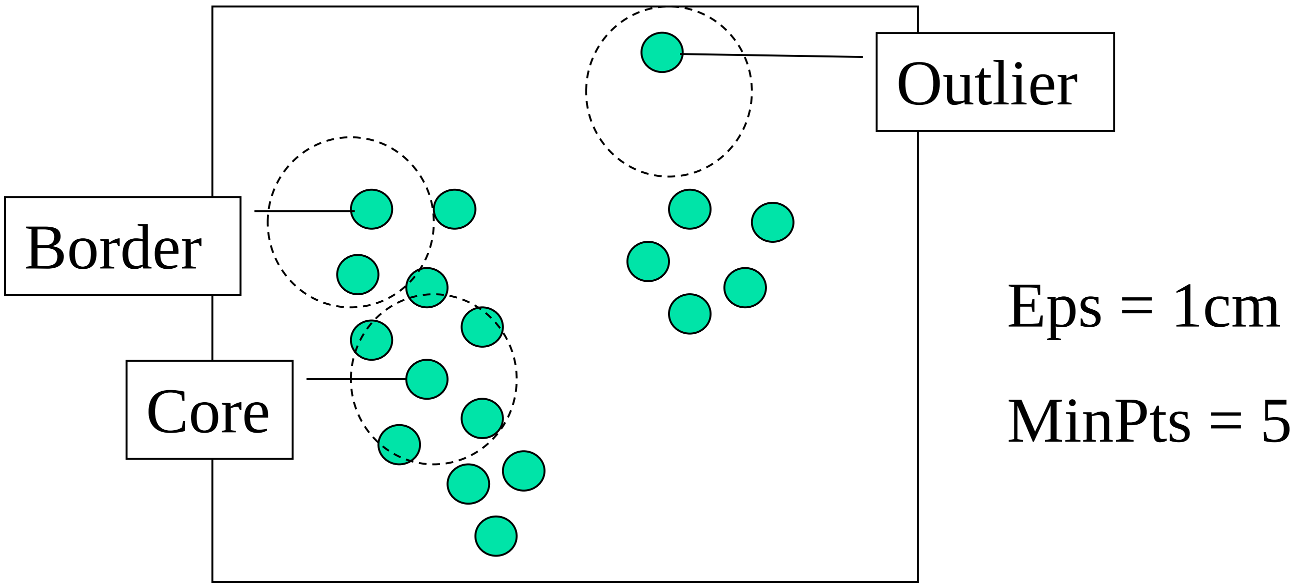


- Density-connected
  - A point p is density-connected to a point q w.r.t. *Eps* and *MinPts* if there is a point o such that both, p and q are density-reachable from o w.r.t. *Eps* and *MinPts*



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 an *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.

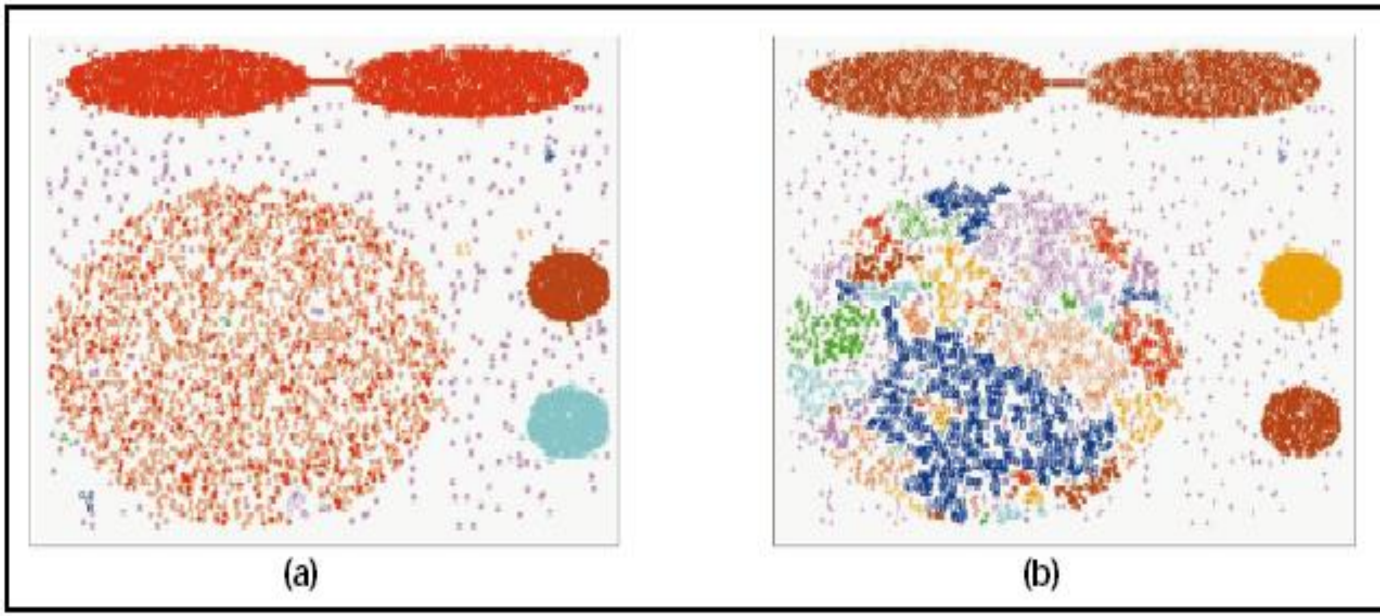
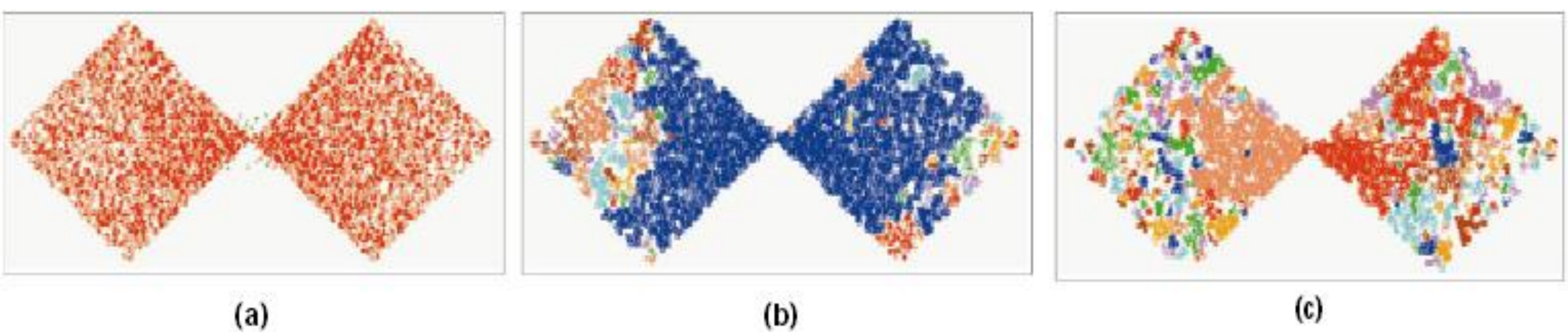
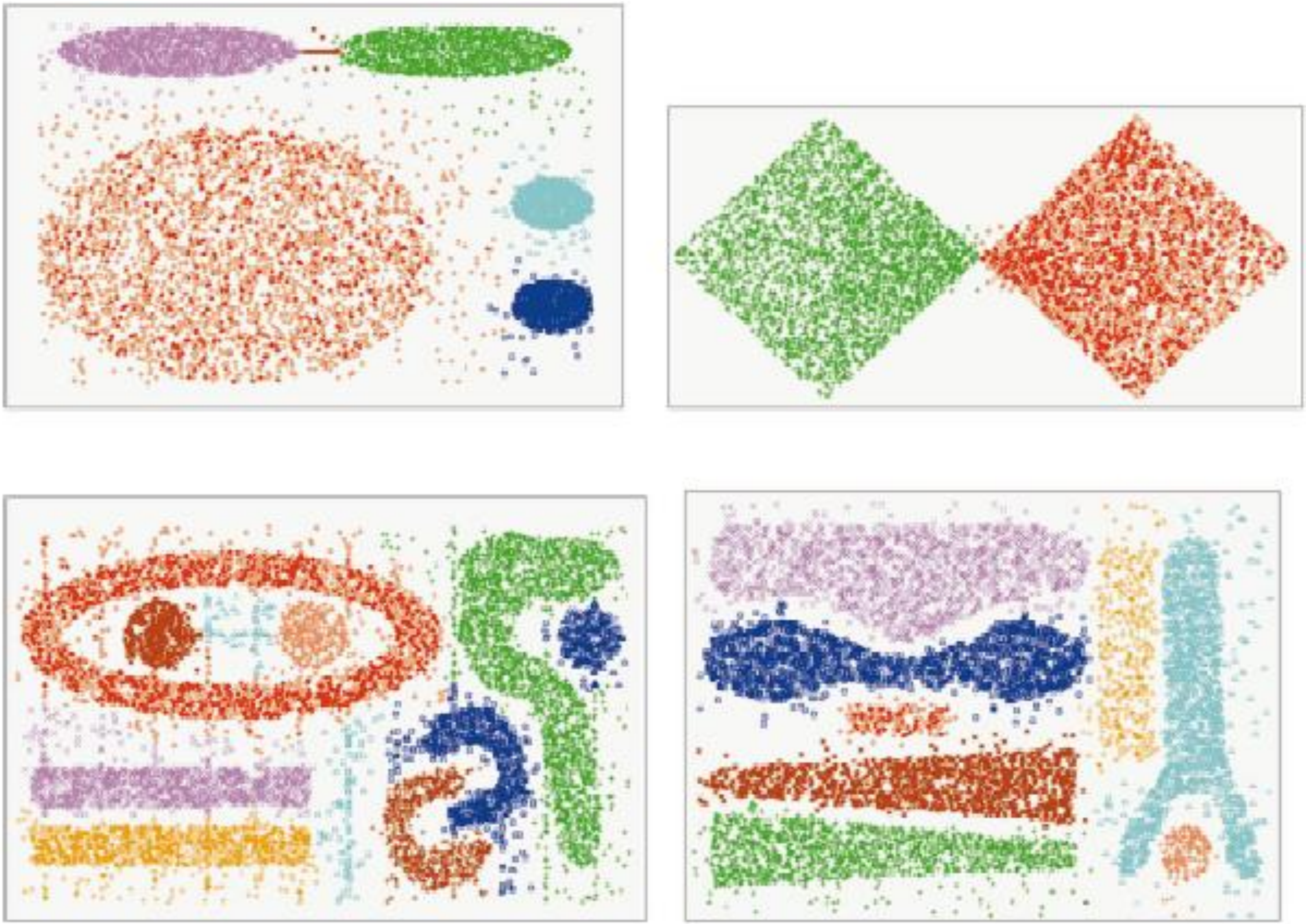


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



CHAMELEON (Clustering Complex Objects)



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 objects in the database wrt its density-based clustering structure
  - This cluster-ordering contains info equiv to different density-based clusterings corresponding to a broad range of parameter settings (*Eps*)
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically or using visualization techniques

OPTICS: A Cluster-Ordering Method (1999)

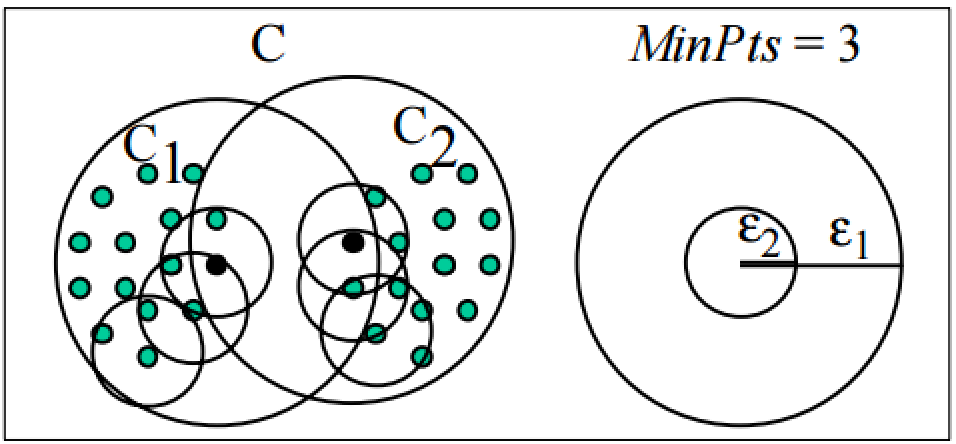


Figure 3. Illustration of "nested" density-based clusters

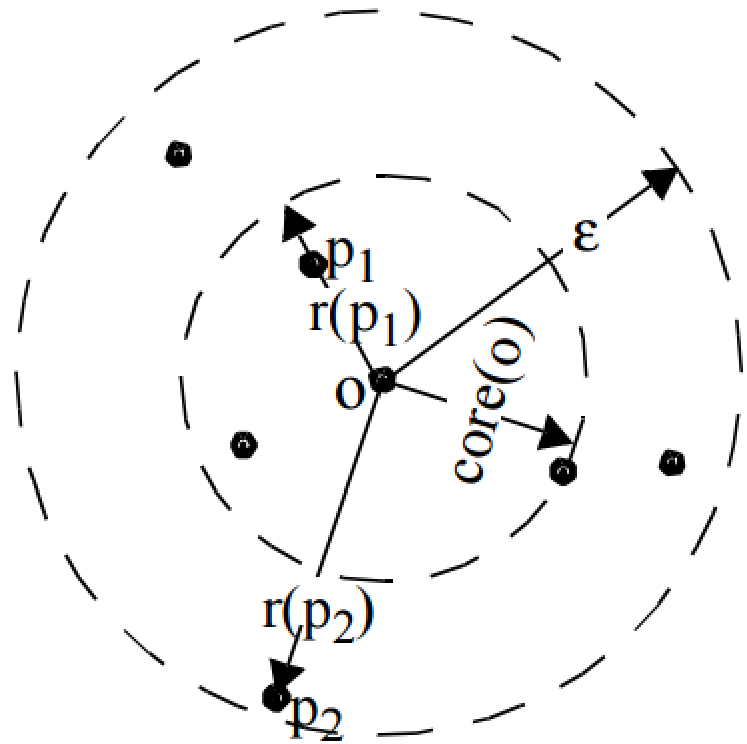
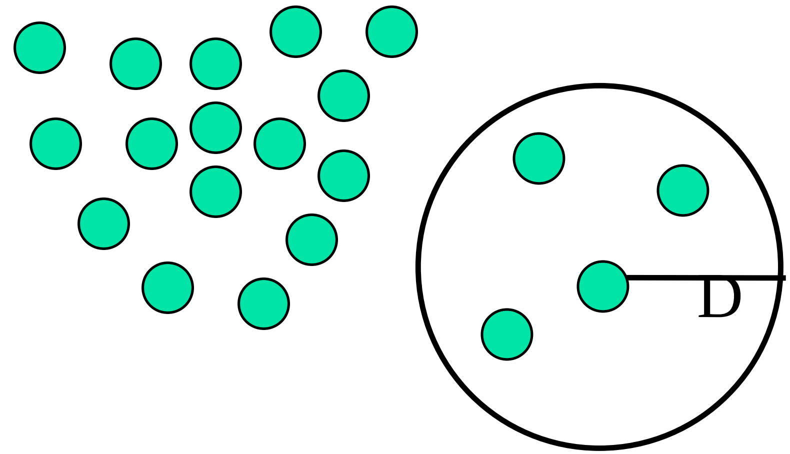


Figure 4. Core-distance(*o*), reachability-distances r(p_1,*o*), r(p_2,*o*) for *MinPts*=4

OPTICS: Some Extension from DBSCAN

- Index-based:
  - k = number of dimensions
  - N = 20
  - p = 75%
  - M = N(1-p) = 5
- Complexity: $O(kN^2)$



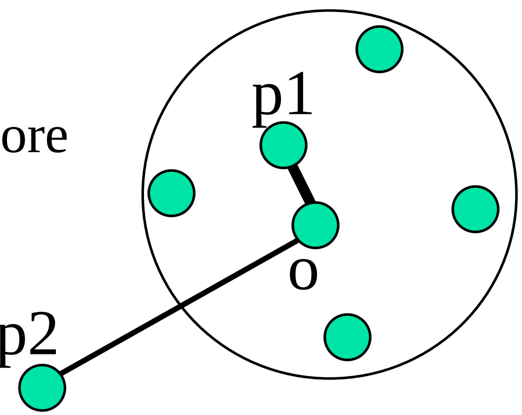
- Core Distance

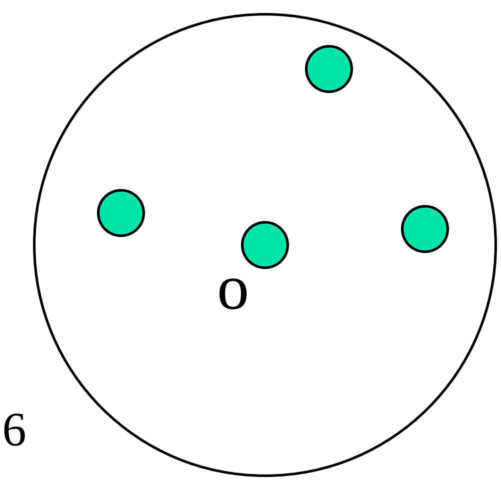
  Distance to make the object a core

- Reachability Distance

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

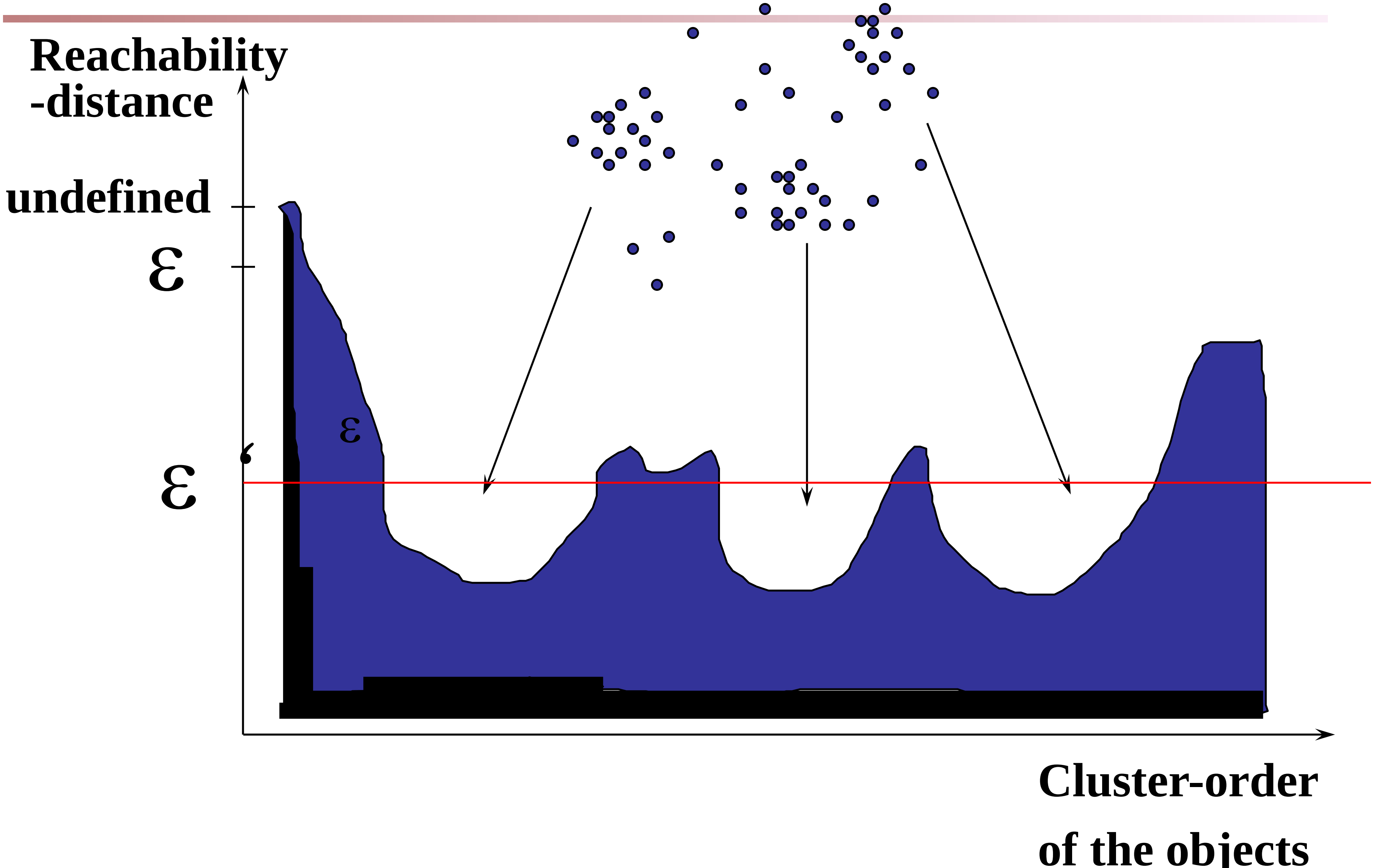
  r(p1, o) = 2.8cm. r(p2,o) = 4cm



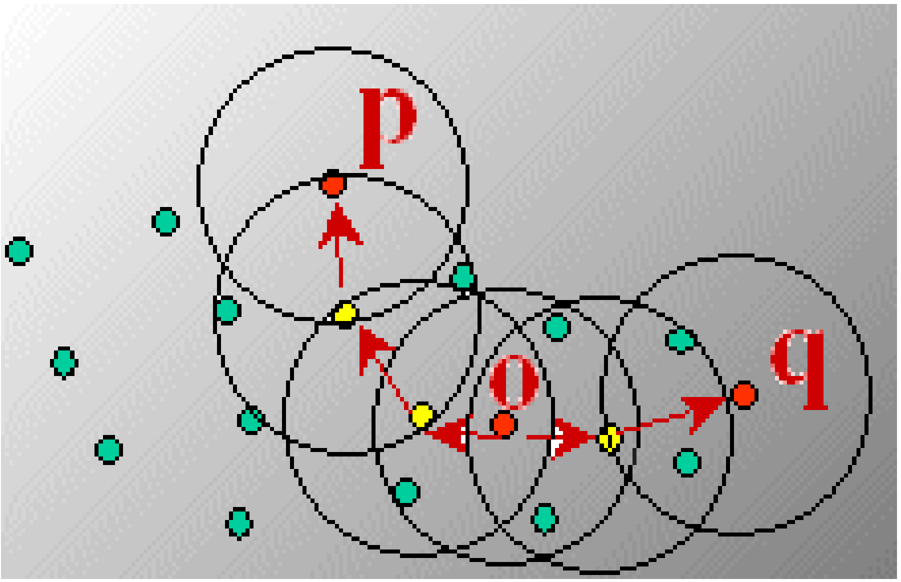


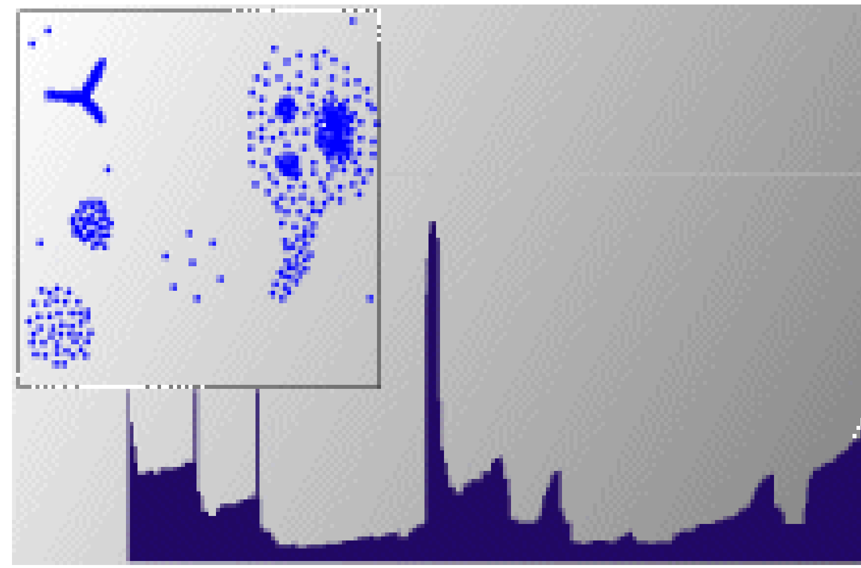
MinPts = 6

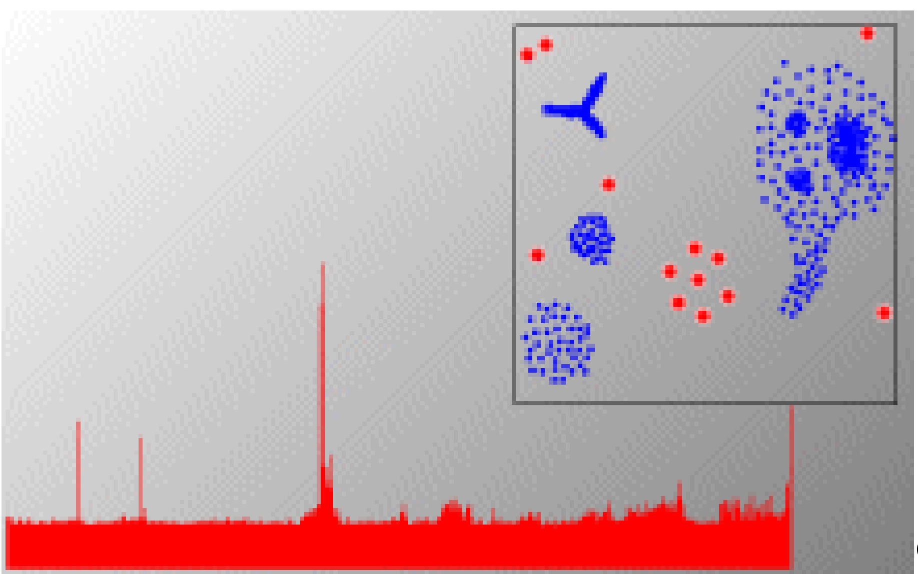
ε = 3 cm

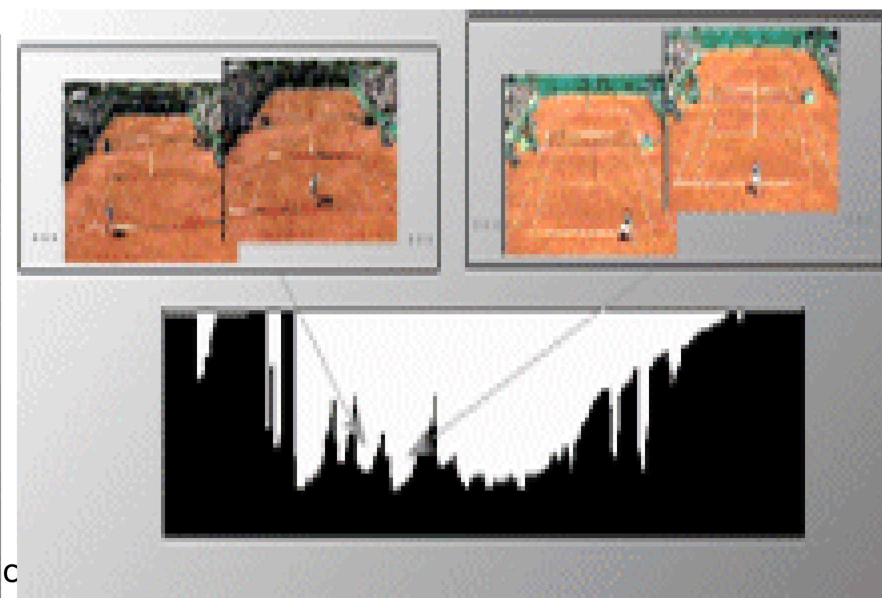

Reachability
-distance
undefined
ε
ε
‘
ε
Cluster-order
of the objects

Density-Based Clustering: OPTICS & Its Applications









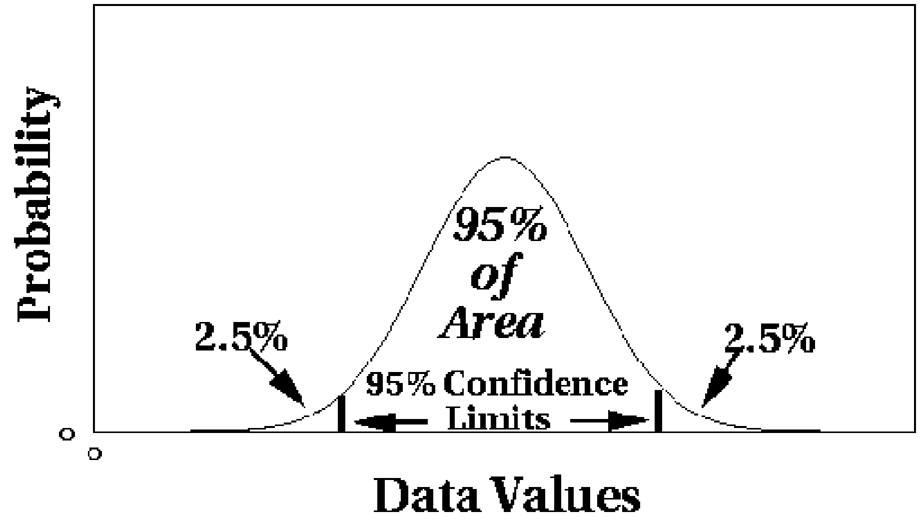
Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. Summary

What Is Outlier Discovery?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Example: Sports: Michael Jordon, Wayne Gretzky, ...
- Problem: Define and find outliers in large data sets
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis

Outlier Discovery: Statistical Approaches



Assume a model underlying distribution that generates data set (e.g. normal distribution)

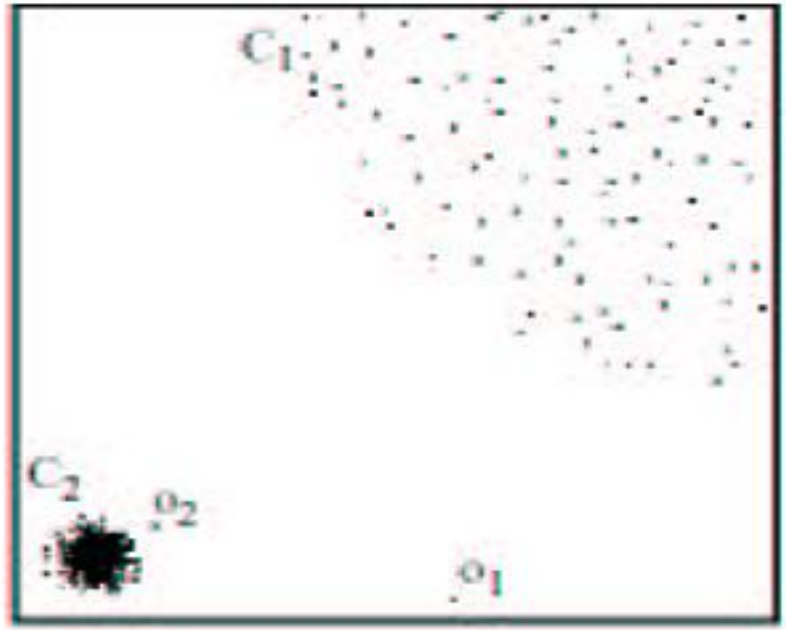
- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers
- Drawbacks
  - most tests are for a *single attribute*
  - In many cases, data distribution may not be known

Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution
- Distance-based outlier: A *DB(p, D)-outlier* is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O
- Algorithms for mining distance-based outliers
  - Index-based algorithm
  - Nested-loop algorithm
  - Cell-based algorithm

Density-Based Local Outlier Detection

- Distance-based outlier detection is based on global distance distribution
- It encounters difficulties to identify outliers *if data is not uniformly distributed*
- Ex. C_1 contains 400 loosely distributed points, C_2 has 100 tightly condensed points, 2 outlier points o_1, o_2
- Distance-based method cannot identify o_2 as an outlier
- Need the concept of *a local outlier*



- Local outlier factor (LOF)
  - Assume outlier is not crisp
  - Each point has a LOF

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. 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
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis

Problems and Challenges

- Considerable progress has been made in scalable clustering methods
  - Partitioning: k-means, k-medoids, CLARANS
  - Hierarchical: BIRCH, ROCK, CHAMELEON
  - Density-based: DBSCAN, OPTICS, DenClue
  - Constraint-based: COD, constrained-clustering
- Current clustering techniques do not address all the requirements adequately, still an active area of research