# Data Mining: Foundation, Techniques and Applications

## Lesson 7,8:Clustering and Outlier Detection

Li Cuiping(李翠平)
School of Information
Renmin University of China

Anthony Tung(鄧锦浩)
School of Computing
National University of Singapore

# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- 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
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

# General Applications of Clustering

- Pattern Recognition
- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

# Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then 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
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

# What Is Good Clustering?

- A <u>good clustering</u> method will produce high quality clusters with
  - high <u>intra-class</u> similarity
  - low <u>inter-class</u> similarity
- The <u>quality</u> of a clustering result depends on both the similarity measure used by the method and its implementation.
- The <u>quality</u> of a clustering method is also measured by its ability to discover some or all of the <u>hidden</u> patterns.

# Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

# Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

# Data Structures

- Data matrix
  - (two modes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

# Measure the Quality of Clustering

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

# Type of data in clustering analysis

- Interval-scaled variables:
- Binary variables:
- Nominal, ordinal, and ratio variables:
- Variables of mixed types:

# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

# Interval-valued variables

- Standardize data
  - Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf})$.

  - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

# Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two $p$-dimensional data objects, and $q$ is a positive integer

- If $q = 1$, $d$ is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

# Similarity and Dissimilarity Between Objects (Cont.)

- *If q = 2, d* is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

  - Properties
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$
- Also one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures.

# Binary Variables

- A contingency table for binary data

| | | Object $j$ | | |
|---|---|---|---|---|
| | | 1 | 0 | *sum* |
| | 1 | $a$ | $b$ | $a+b$ |
| **Object $i$** | 0 | $c$ | $d$ | $c+d$ |
| | *sum* | $a+c$ | $b+d$ | $p$ |

- Simple matching coefficient (invariant, if the binary variable is *symmetric*): $d(i, j) = \frac{b + c}{a + b + c + d}$
- Jaccard coefficient (noninvariant if the binary variable is *asymmetric*): $d(i, j) = \frac{b + c}{a + b + c}$

# Dissimilarity between Binary Variables

- Example

| Name | Gender | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|---|---|---|---|---|---|---|---|
| Jack | M | Y | N | P | N | N | N |
| Mary | F | Y | N | P | N | P | N |
| Jim | M | Y | P | N | N | N | N |

  - gender is a symmetric attribute
  - the remaining attributes are asymmetric binary
  - let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

# Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

# Ordinal Variables

- An ordinal variable can be discrete or continuous
- order is important, e.g., rank
- Can be treated like interval-scaled
  - replacing $x_{if}$ by their rank $r_{if} \in \{1, \ldots, M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables

# Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale,
  such as $Ae^{Bt}$ or $Ae^{-Bt}$
- Methods:
  - treat them like interval-scaled variables — *not a good choice! (why?)* *Example: Difference between 0.5 and 1.0 could be less significant than difference between 0.0 to 0.1*
  - apply logarithmic transformation

$$y_{if} = log(x_{if})$$

  - treat them as continuous ordinal data treat their rank as interval-scaled.

# Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio.
- One may use a weighted formula to combine their effects.

$$d(i,j) = \frac{\Sigma_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\Sigma_{f=1}^{p} \delta_{ij}^{(f)}}$$

  - $f$ is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$ , or $d_{ij}^{(f)} = 1$ o.w.
  - $f$ is interval-based: use the normalized distance
  - $f$ is ordinal or ratio-scaled
    - compute ranks $r_{if}$ and
    - and treat $z_{if}$ as interval-scaled $z_{if} = \frac{r_{if} - 1}{M_f - 1}$

# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

# Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure

# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

# Partitioning Algorithms: Basic Concept

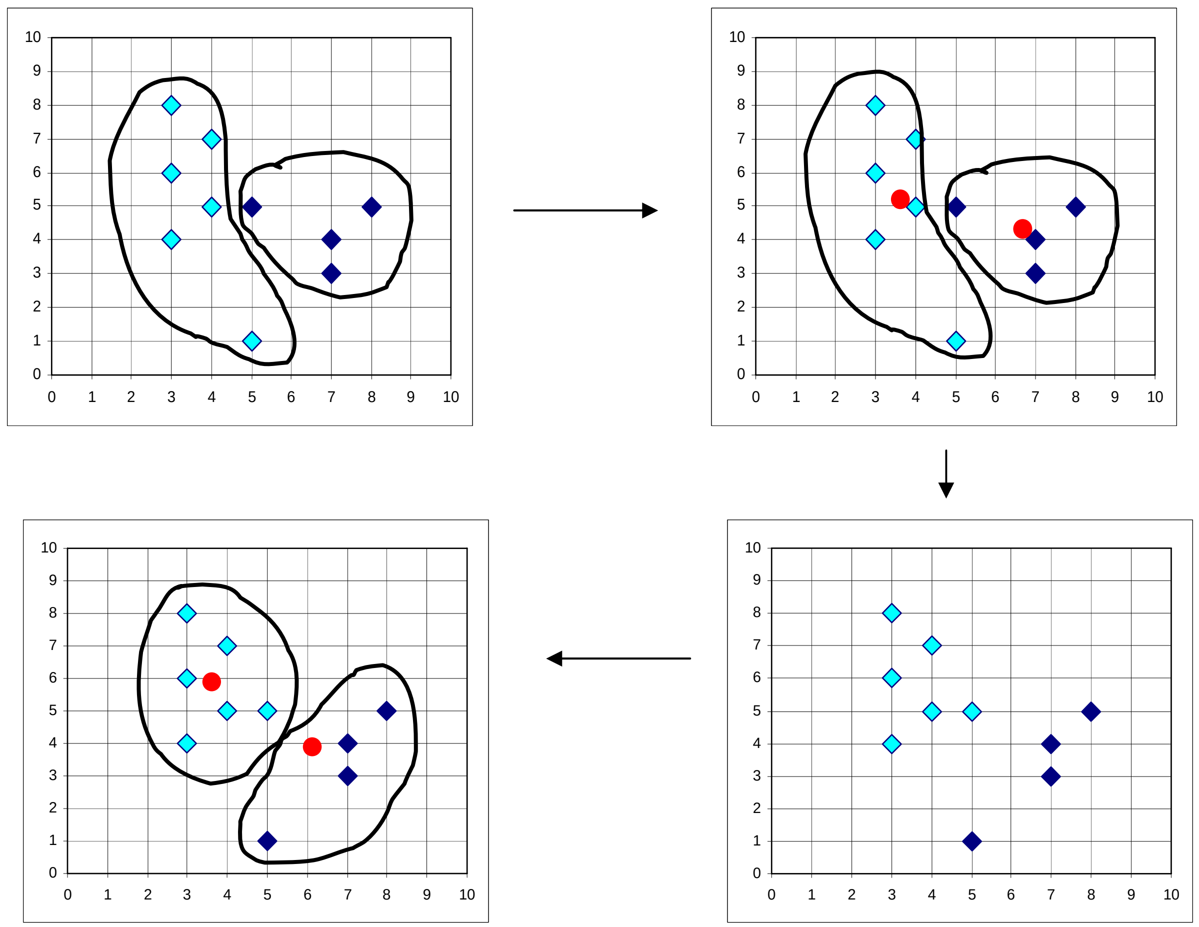
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* 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 4 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 (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*(*tkdn*), where *n* is # objects, *k* is # clusters, d is the # of dimensions and *t* is # iterations. Normally, *k*, *t*, *d* << *n*.
  - Often terminates at a *local optimum*.
- Weakness
  - Need to specify *k*, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

# Can we terminate k-means earlier?

- The k-means algorithm must be ran multiple times to get better result. How do we know a set of initial centers will not give better result?
- Compute a bound on how much can future iterations improve on the objective function. If it is too small, terminate at once.
  - Zhenjie Zhang, Bing Tian Dai and Anthony K.H. Tung. "On the Lower Bound of Lower Optimums in K-Means Algorithm". In ICDM 2006. [Codes][PPT]

# The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 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
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling

# *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
- <u>Strength</u>: deals with larger data sets than *PAM*
- <u>Weakness:</u>
  - 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)*

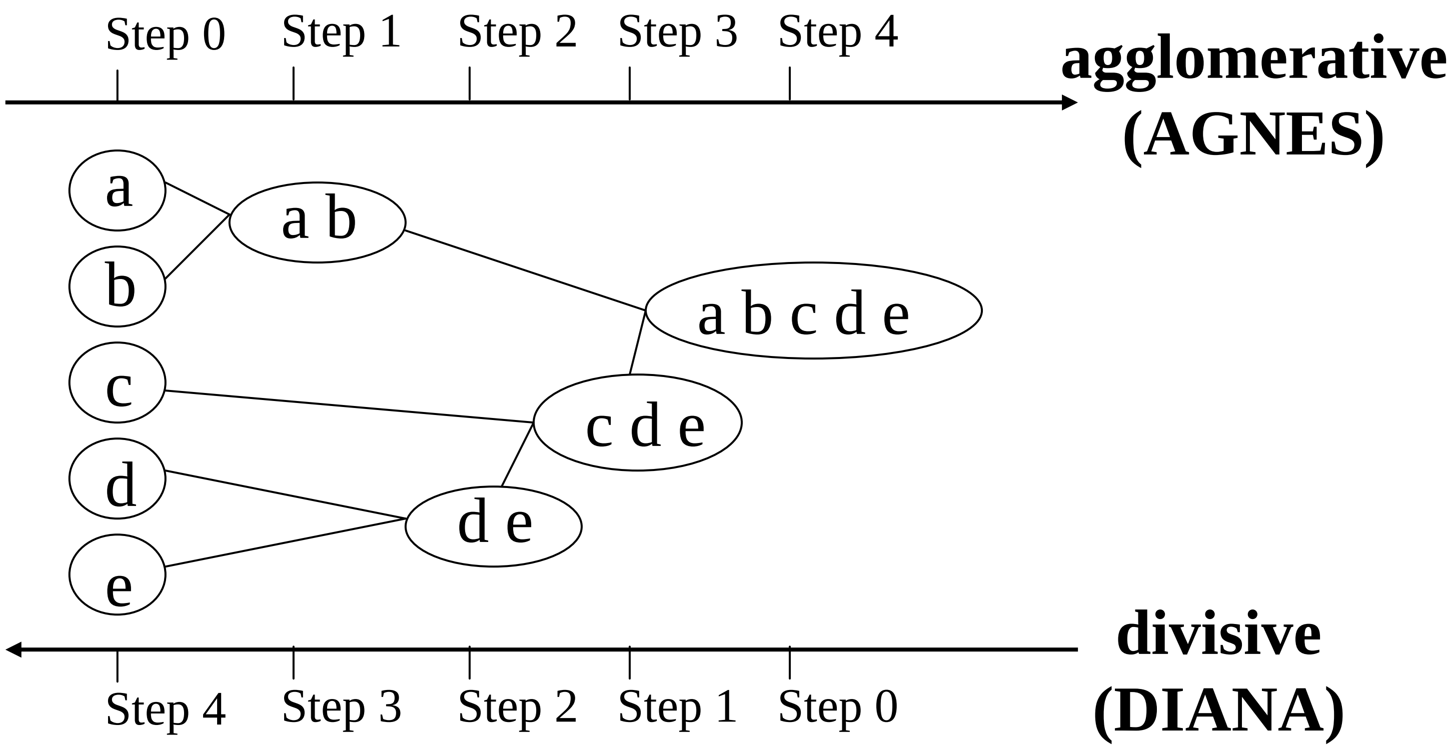
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS*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

# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

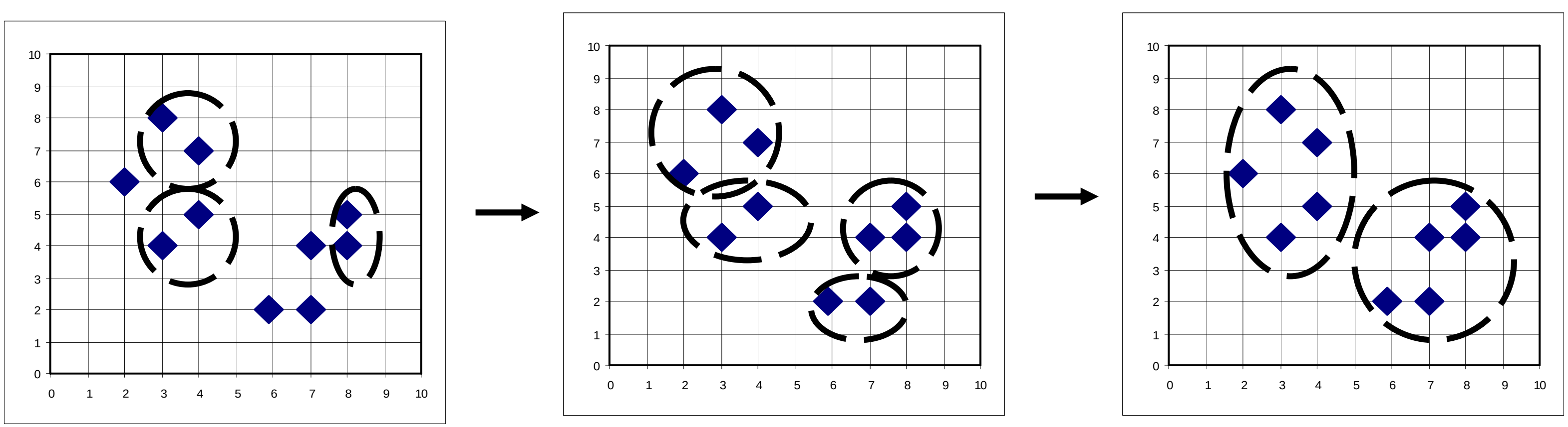
# Hierarchical Clustering

- 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)

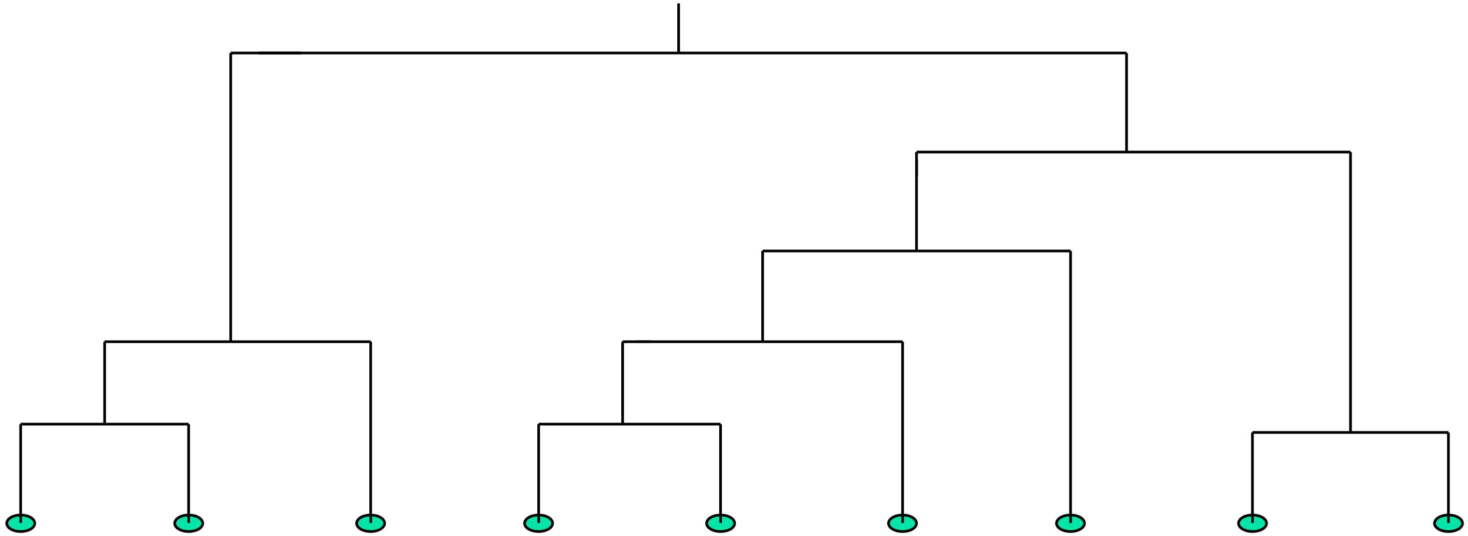
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



# A *Dendrogram* Shows How the Clusters are Merged Hierarchically

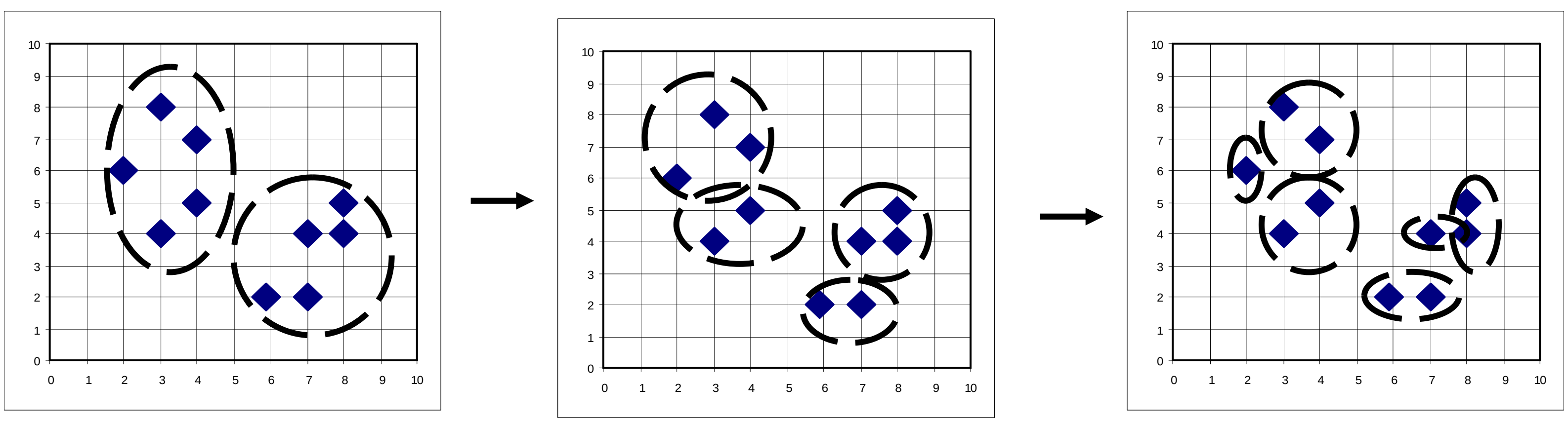
**Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.**

**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, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



# More on 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
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

# BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by 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:* sensitive to the order of the data record.

# Clustering Feature Vector

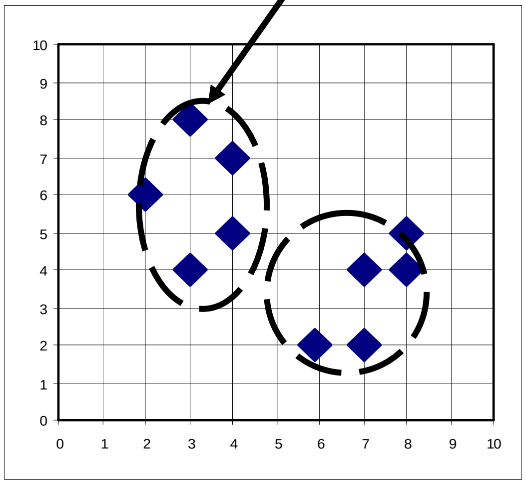
**Clustering Feature:** $\boldsymbol{CF = (N, LS, SS)}$

$N$: **Number of data points**

$LS: \sum^{N}_{i=1} = \overrightarrow{X_i}$

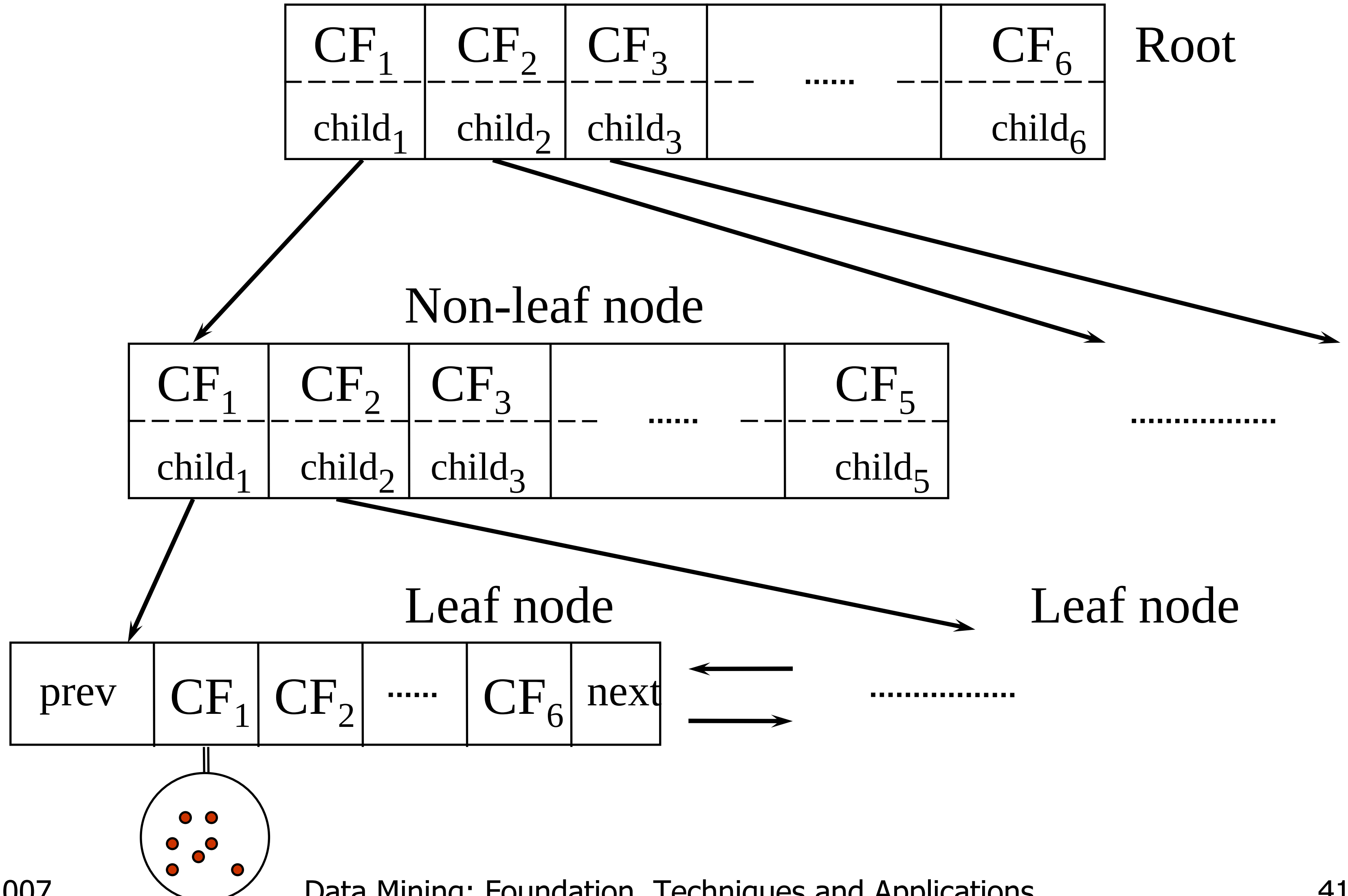
$SS: \sum^{N}_{i=1} = \overrightarrow{X_i}^{2}$

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

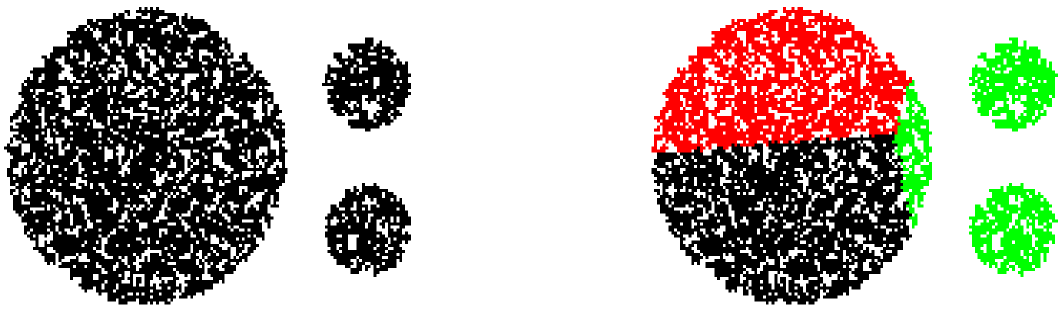



(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

# CF Tree


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

# CURE (Clustering Using REpresentatives )

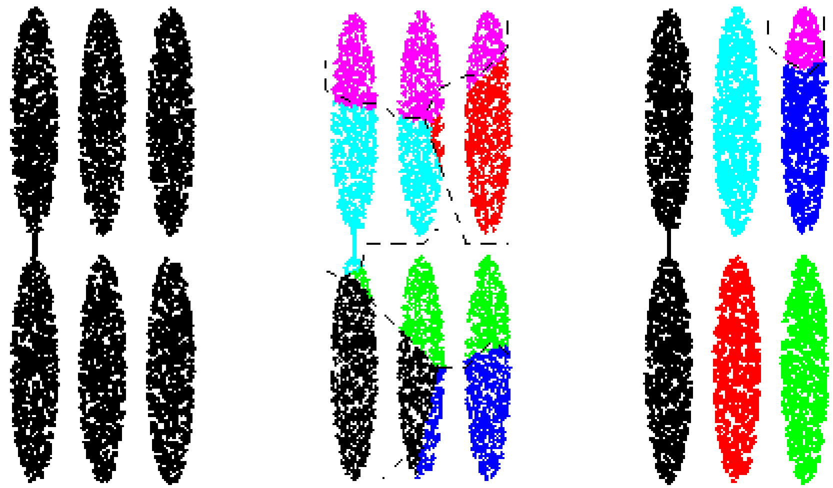


(a)

(b)

- CURE: proposed by Guha, Rastogi & Shim, 1998
  - Stops the creation of a cluster hierarchy if a level consists of *k* clusters
  - Uses multiple representative points to evaluate the distance between clusters, adjusts well to arbitrary shaped clusters and avoids single-link effect

# Drawbacks of Distance-Based Method



(a) (b) (c)

- Drawbacks of square-error based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex shaped, similar size and density, and if *k* can be reasonably estimated

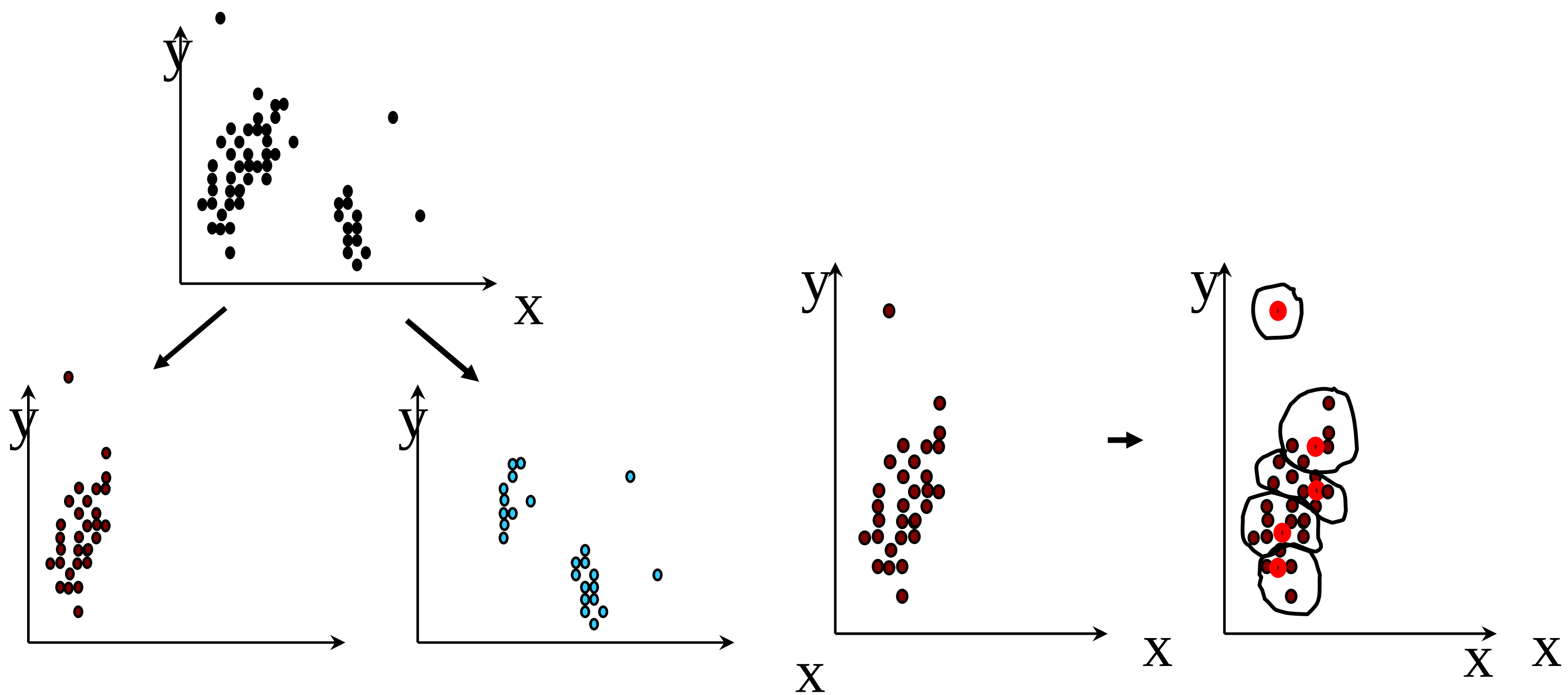
# Cure: The Algorithm

- Draw random sample *s*.
- Partition sample to *p* partitions with size *s/p*
- Partially cluster partitions into *s/pq* clusters
- Eliminate outliers
  - By random sampling
  - If a cluster grows too slow, eliminate it.
- Cluster partial clusters.
- Label data in disk

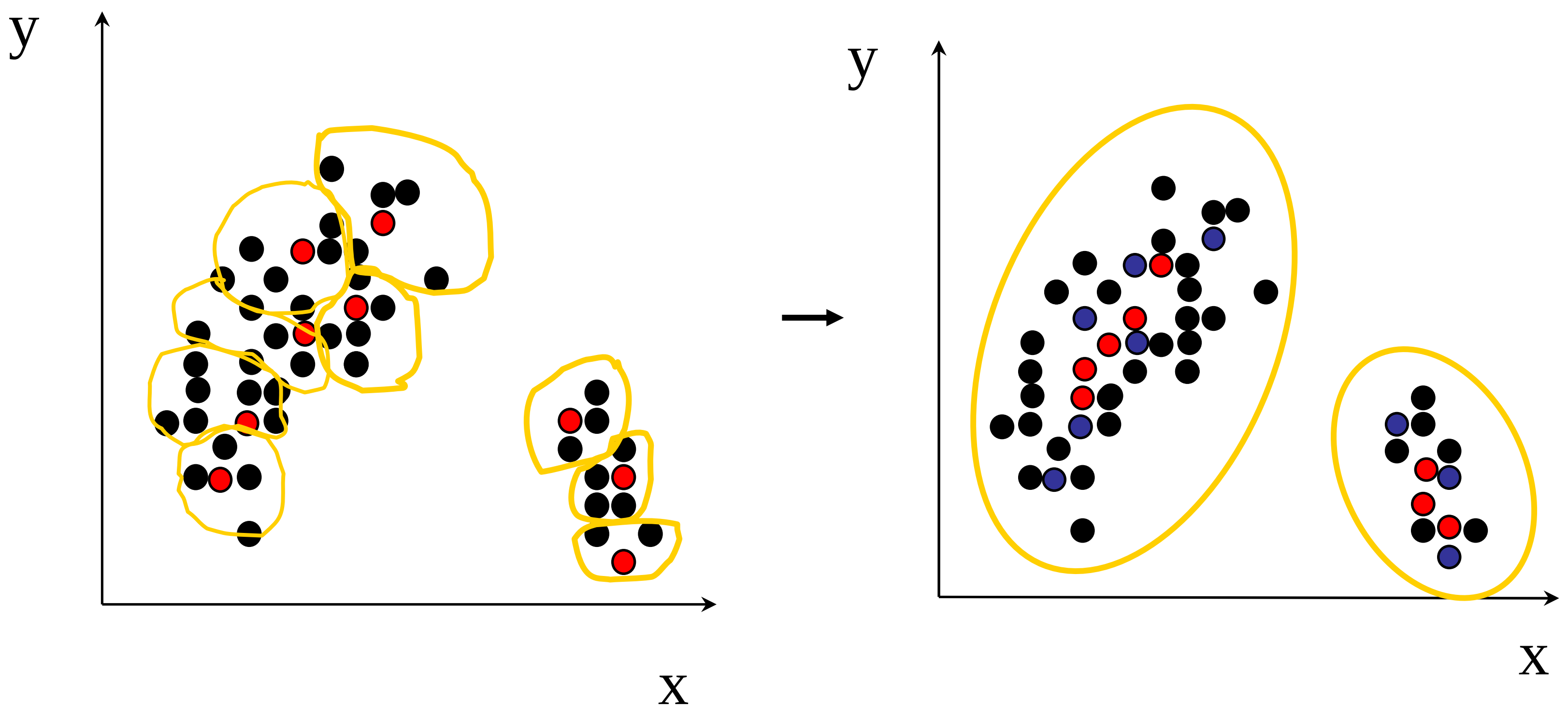
# Data Partitioning and Clustering

- s = 50
- p = 2
- s/p = 25

- s/pq = 5



# Cure: Shrinking Representative Points

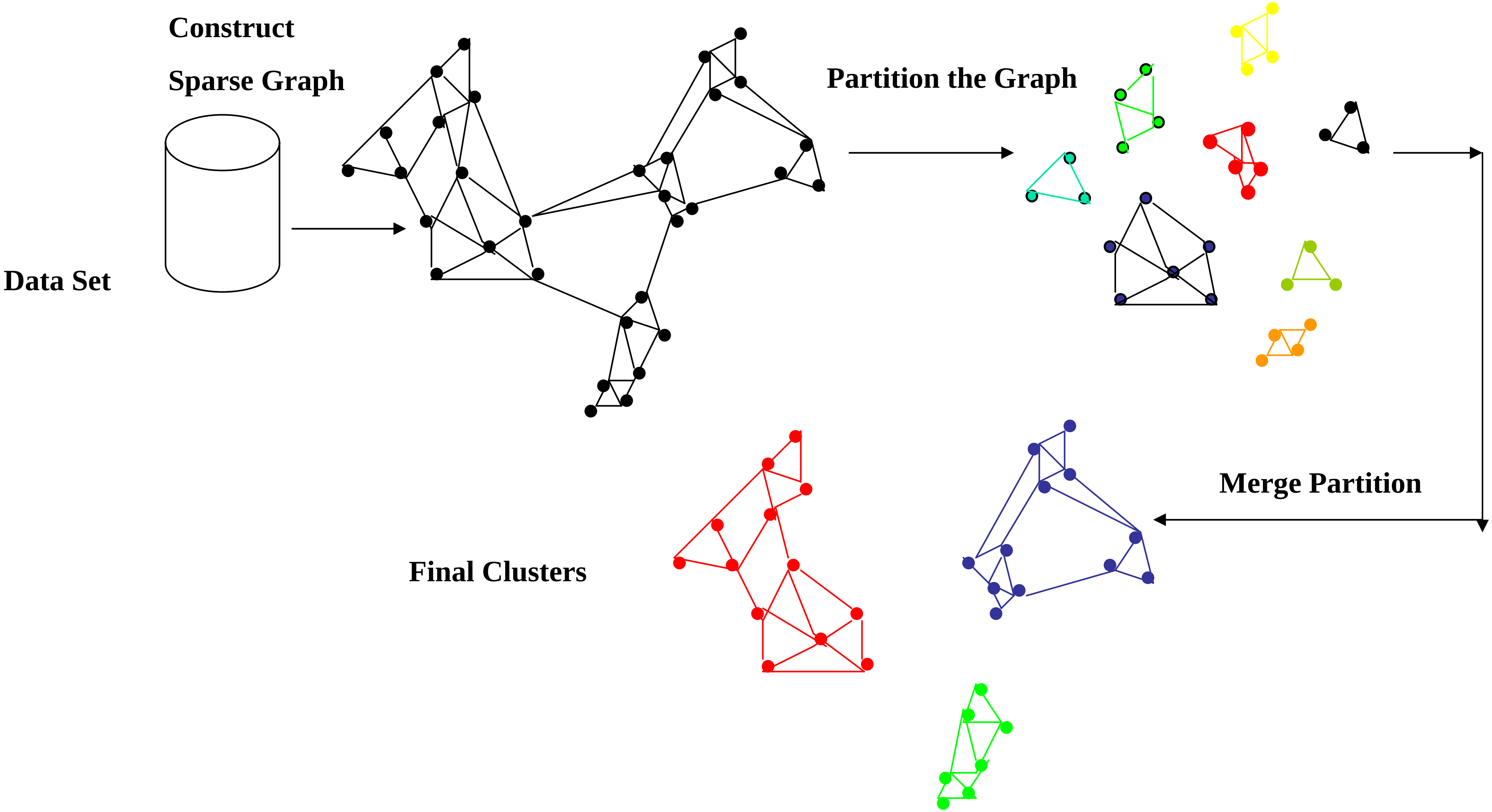



- Shrink the multiple representative points towards the gravity center by a fraction of $\alpha$.
- Multiple representatives capture the shape of the cluster

# CHAMELEON

- CHAMELEON: hierarchical clustering using dynamic modeling, 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 closeness of items within the clusters
- 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



# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

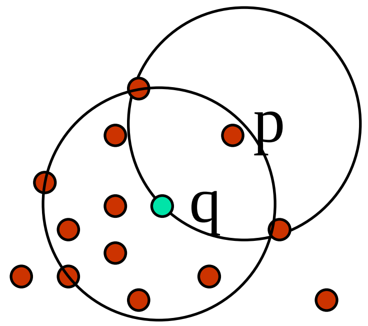
# Density-Based Clustering Methods

- 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)

# Density-Based Clustering: Background

- Two parameters*:*
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that 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*** wrt. ***Eps***, ***MinPts*** if
  - 1) ***p*** belongs to $N_{Eps}(q)$
  - 2) core point condition:

$$|N_{Eps}(q)| >= MinPts$$

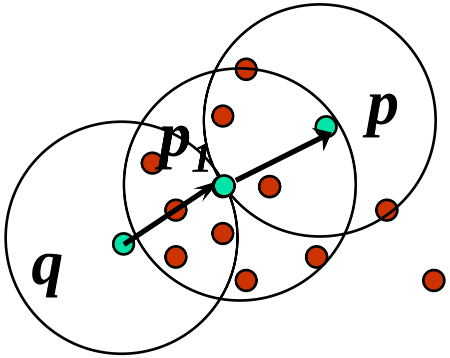



MinPts = 5

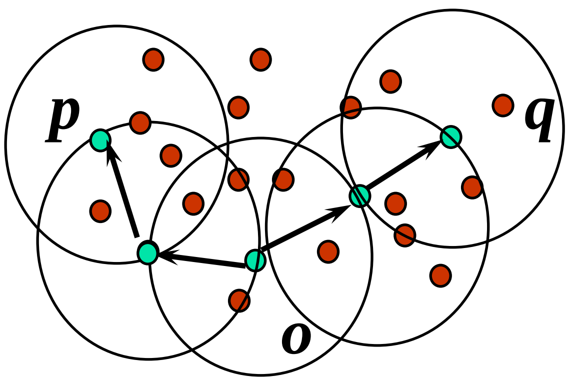
Eps = 1 cm

# Density-Based Clustering: Background (II)

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. *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 $p_i$

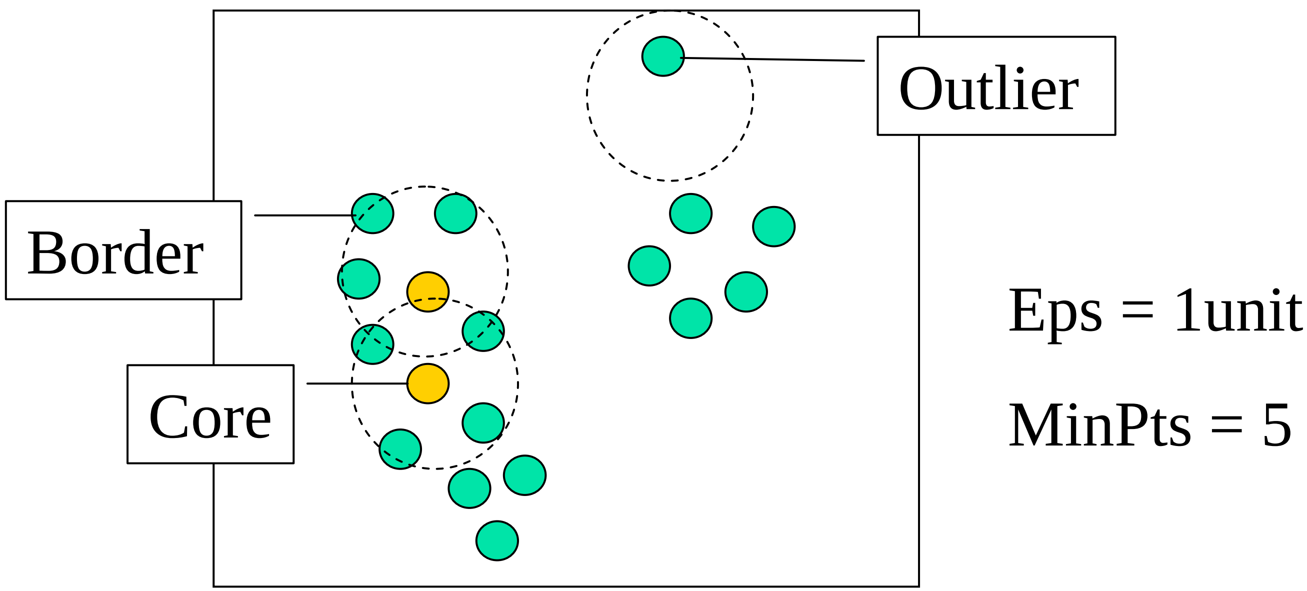



- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



# 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

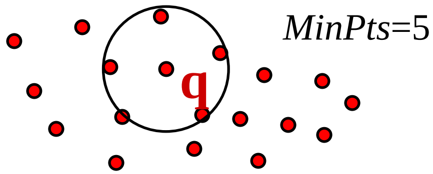
- Arbitrary select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue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

# 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 setting for $\varepsilon' < \varepsilon$ (Note:Eps = $\varepsilon$) and Minpts
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically or using visualization techniques

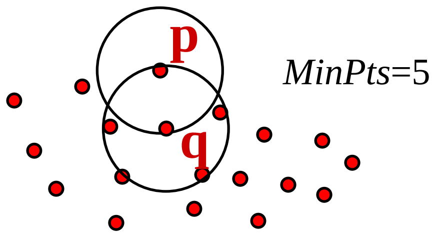
# Density-Based Clustering I

- Parameters
  - range $\varepsilon$ and minimal weight *MinPts*
- Definition: core object
  - *q* is ***core object*** if $|$ *rangeQuery*$(q,\varepsilon)$ $| \geq$ *MinPts*

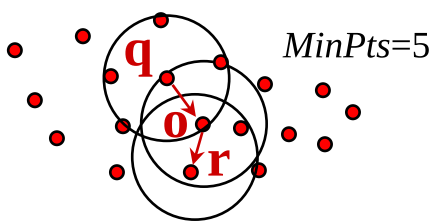



- Definition: directly density-reachable
  - *p* ***directly density-reachable*** from *q* if

    *q* is a core object and $p \in$ *rangeQuery*$(q,\varepsilon)$

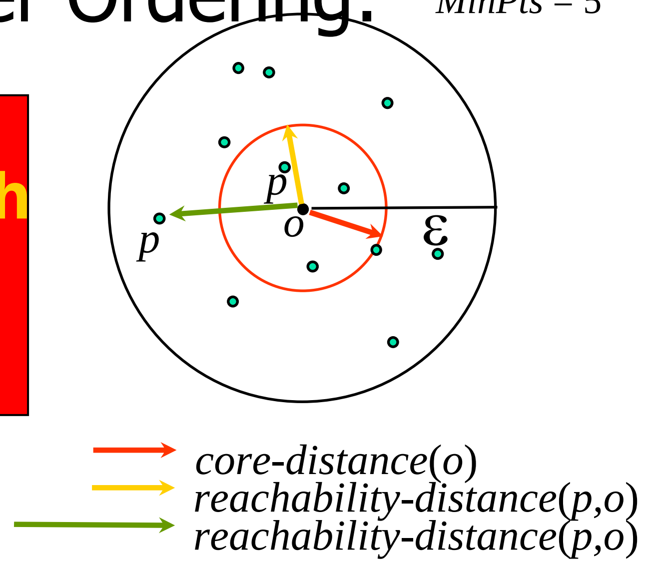



- Definition: density-reachable
  - ***density-reachable***: transitive closure of "directly density-reachable"



# OPTICS

- Core Idea of Hierarchical Cluster Ordering:




**Order the objects linearly such that objects of a cluster are adjacent in the ordering.**

- Definition: core-distance

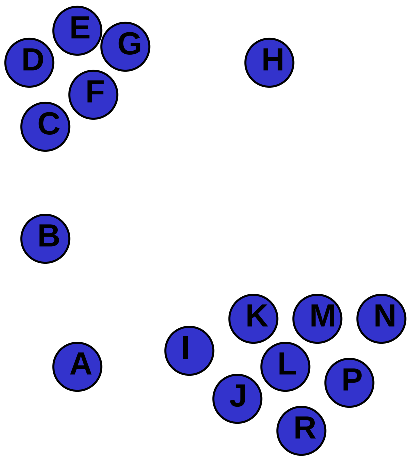
$$\text{core-dist}_{\varepsilon,MinPts}(o) = \begin{cases} \infty & \text{if } \quad |\text{rangeQuery}(o,\varepsilon)| < MinPts \\ MinPts\text{-dist}(o) & \text{otherwise} \end{cases}$$

- Definition: reachability-distance

$$reach\text{-}dist_{\varepsilon,MinPts}(p,o) = max(core\text{-}dist_{\varepsilon,MinPts}(o), dist(p,o))$$

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$

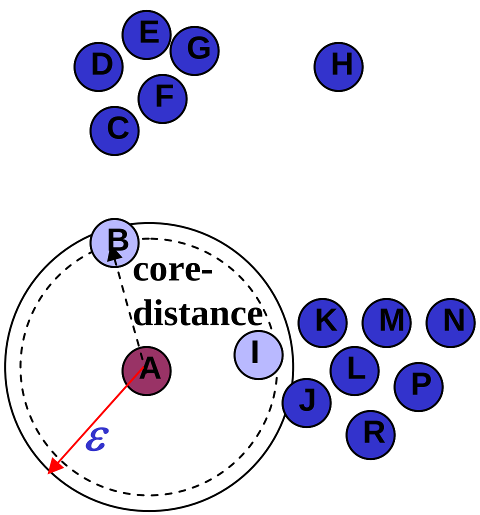


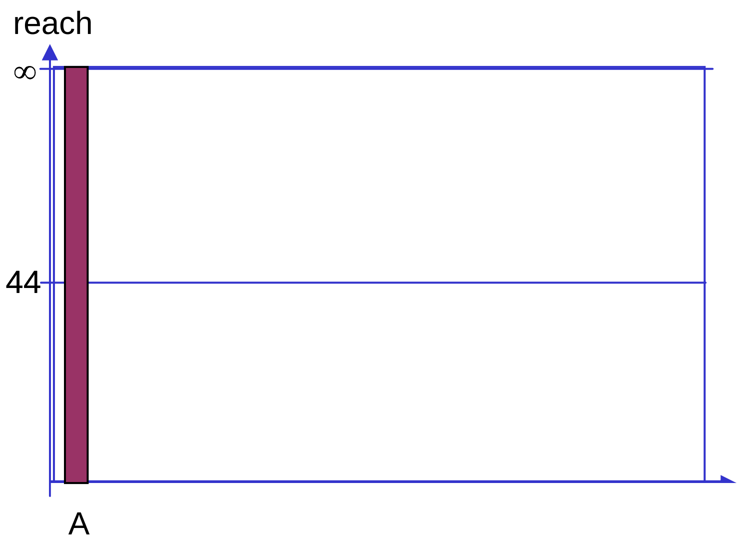


seedlist:

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, *MinPts* = 3

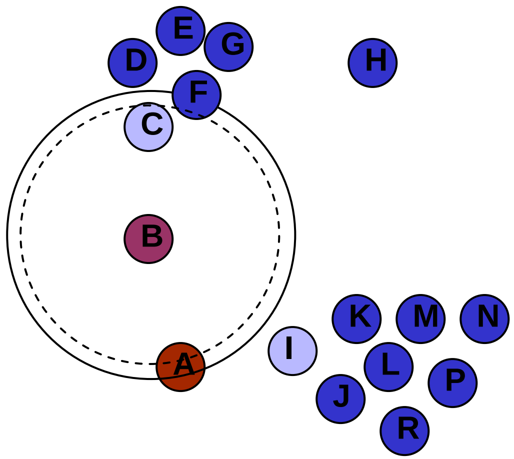


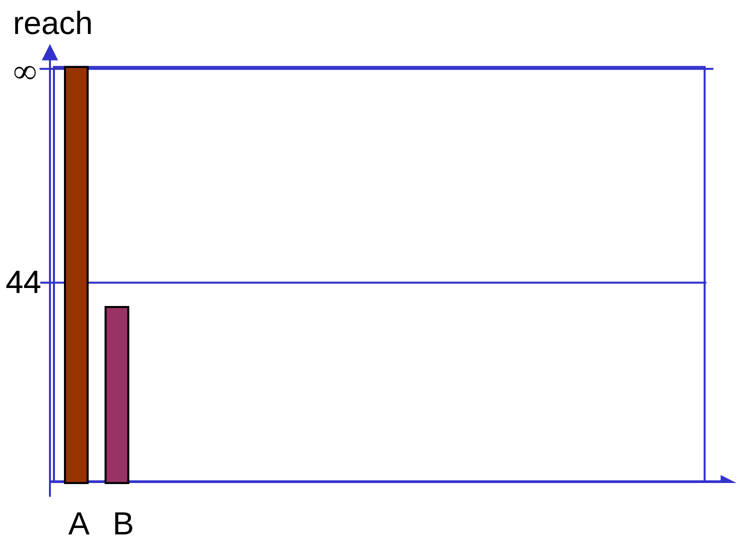




seedlist: (B,40) (I, 40)

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$

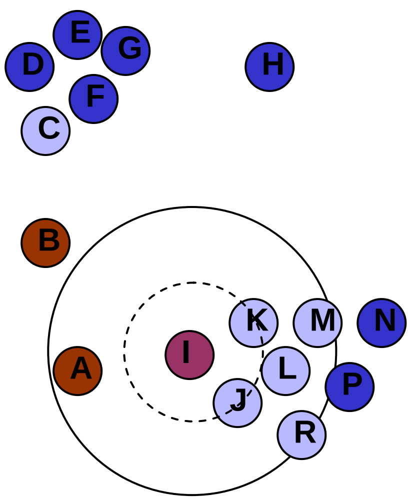


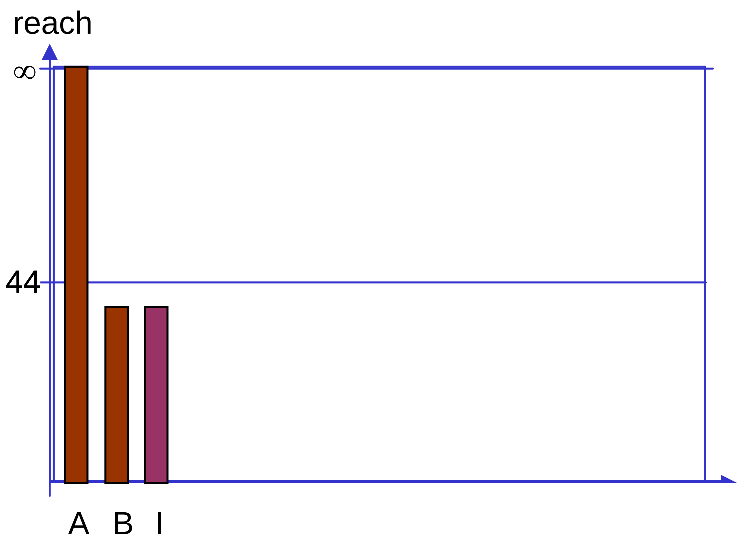




seedlist: (I, 40) (C, 40)

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$

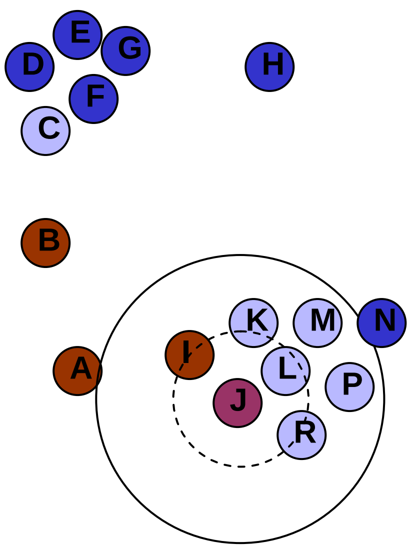


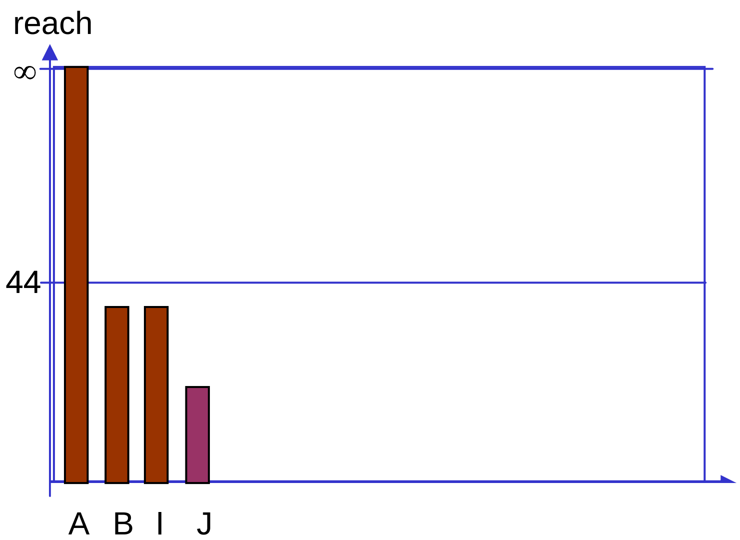




seedlist: (J, 20) (K, 20) (L, 31) (C, 40) (M, 40) (R, 43)

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$

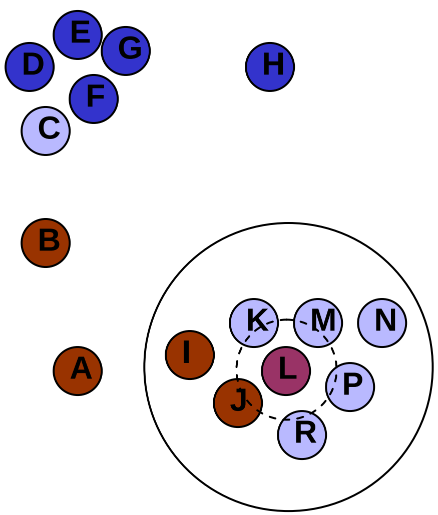


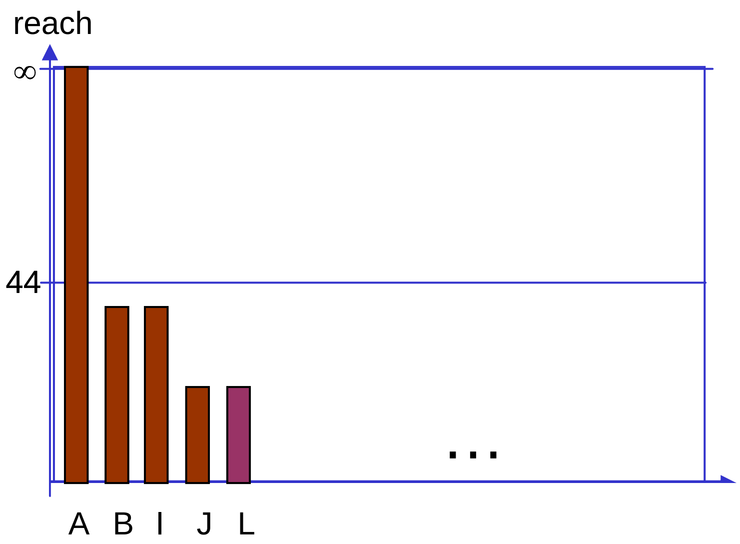




seedlist: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$

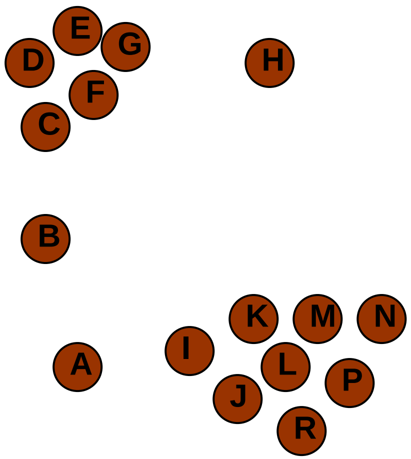


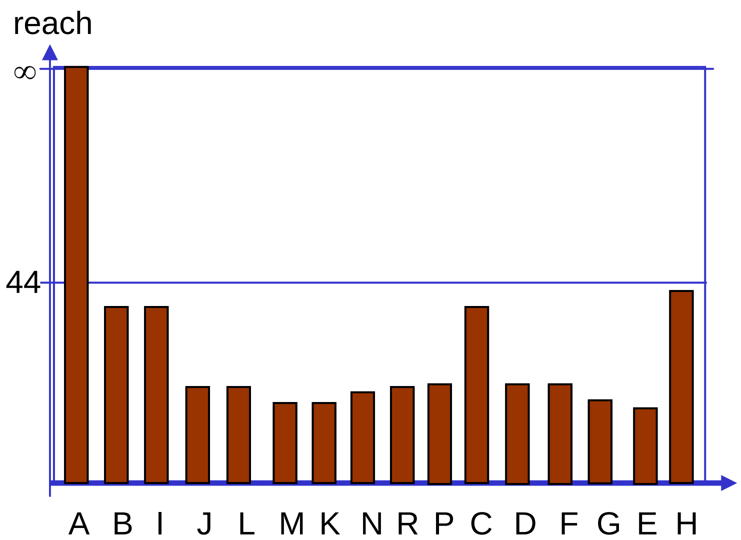




seedlist: (M, 18) (K, 18) (R, 20) (P, 21) (N, 35) (C, 40)

# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$

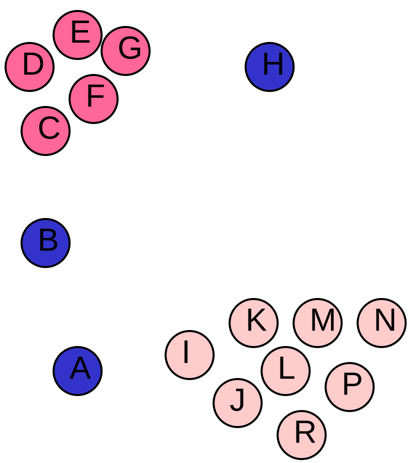


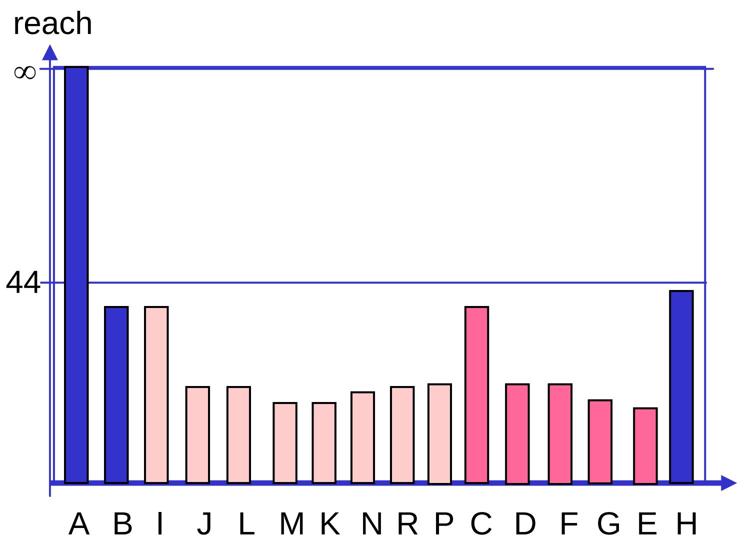




seedlist: -

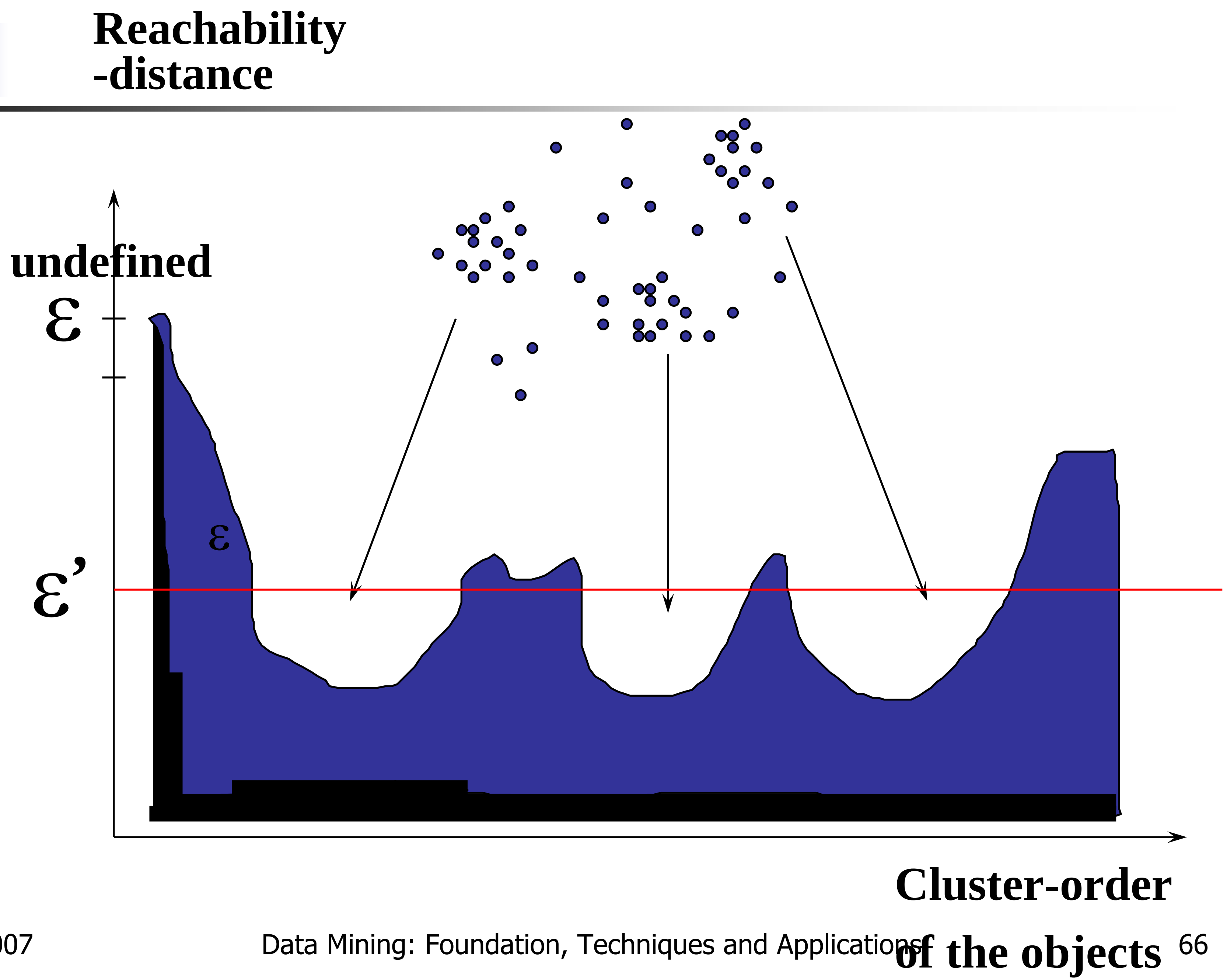
# OPTICS Algorithm

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$




seedlist: -



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

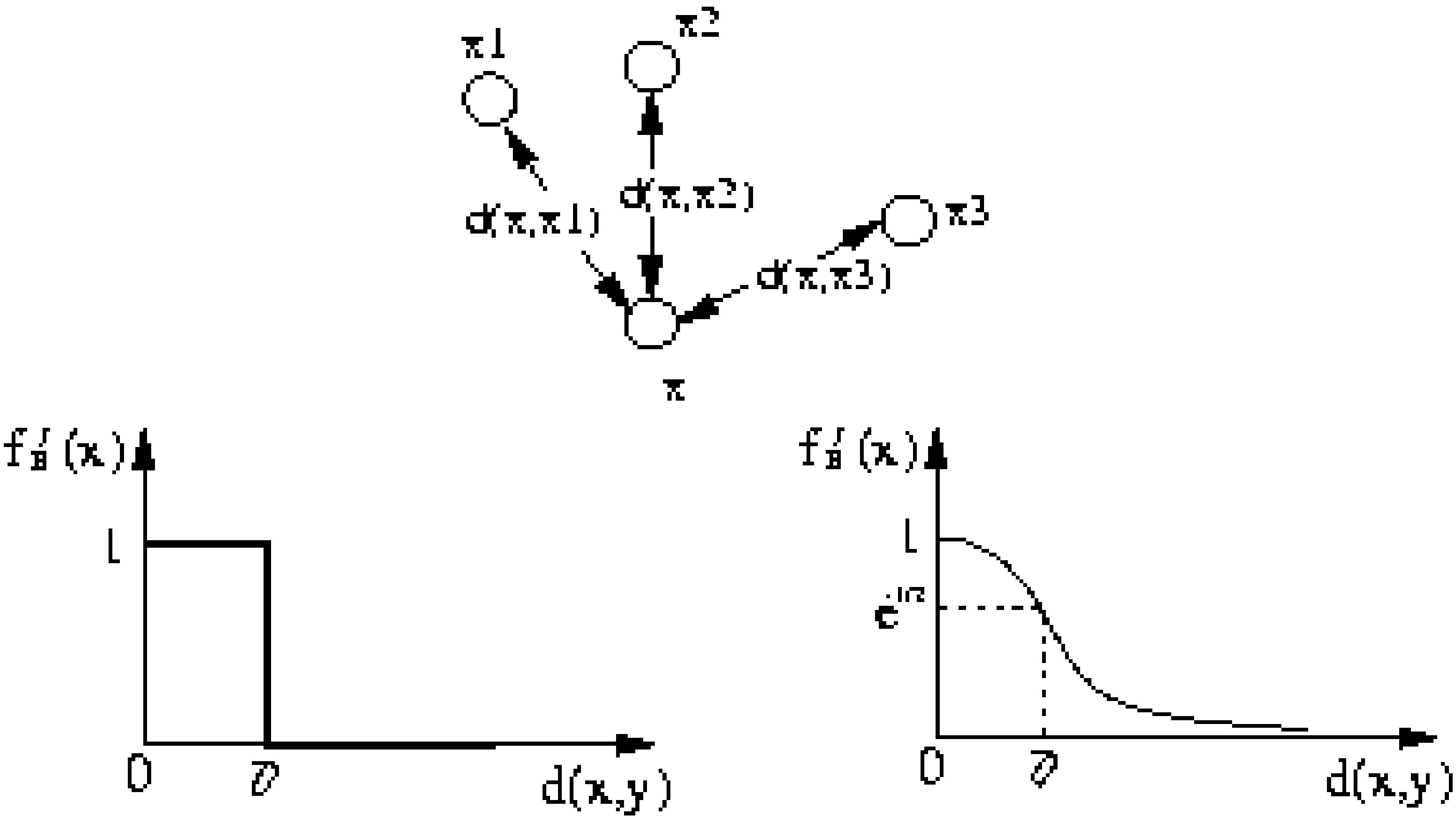
# DENCLUE: using density functions

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
- 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 (faster than DBSCAN by a factor of up to 45)
  - 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.

# Influence Function



# Gradient: The steepness of a slope

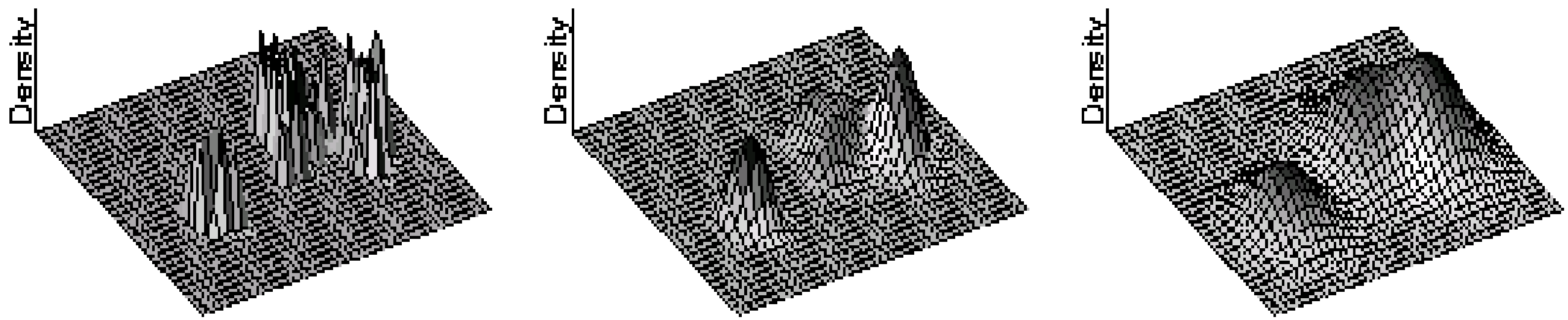
- Example

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

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

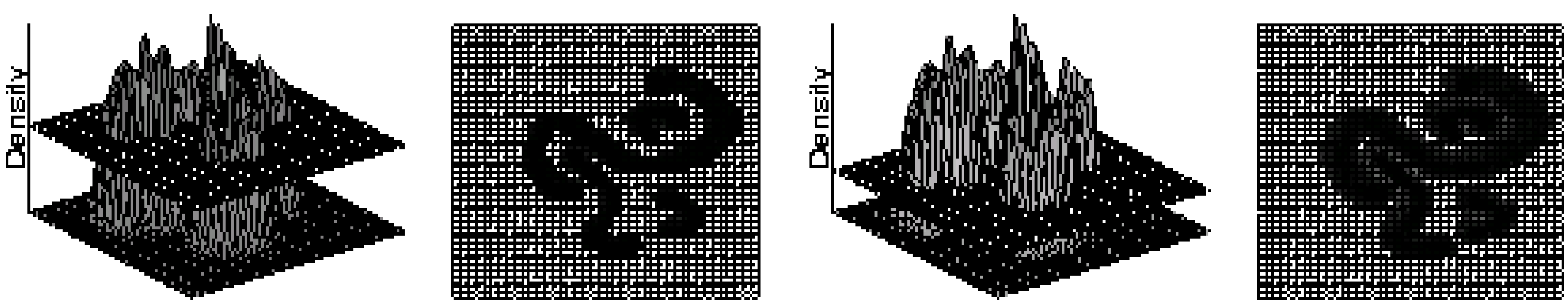
$$\nabla f_{Gaussian}^{D}(x, x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

# 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 $\sigma$




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

**Figure 4:** Example of Arbitray-Shape Clusters for different $\xi$

# Outline

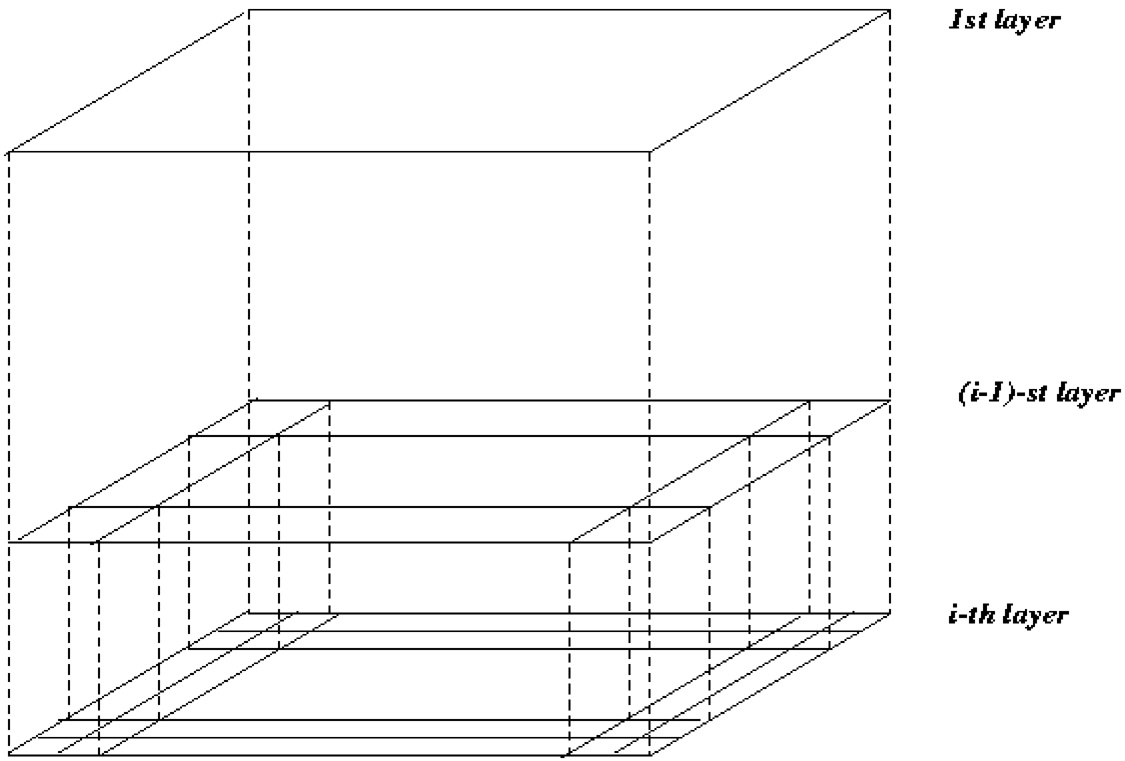
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

# 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)

# STING: A Statistical Information Grid Approach

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

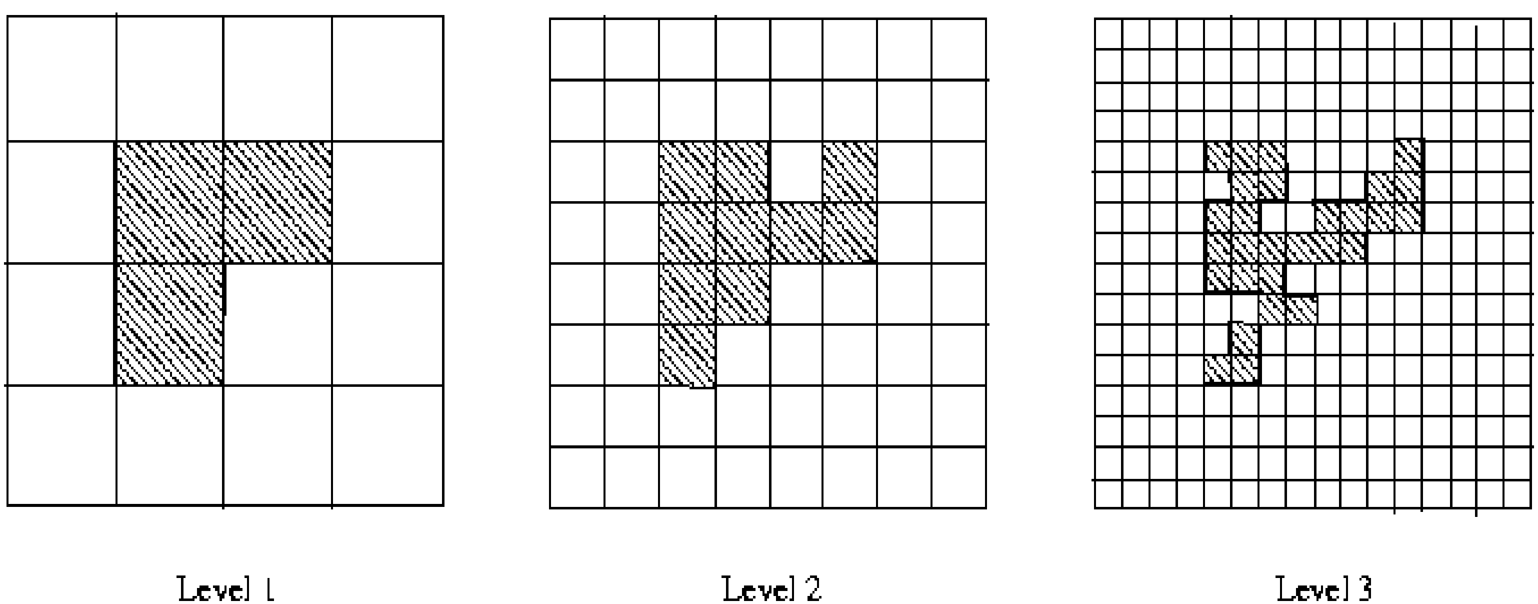


# STING: A Statistical Information Grid Approach (2)

- 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: A Statistical Information Grid Approach (3)

- 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



# WaveCluster (1998)

- Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
- A multi-resolution clustering approach which applies wavelet transform to the feature space
  - A wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band.
- Both grid-based and density-based
- Input parameters:
  - # of grid cells for each dimension
  - the wavelet, and the # of applications of wavelet transform.

# WaveCluster (1998)

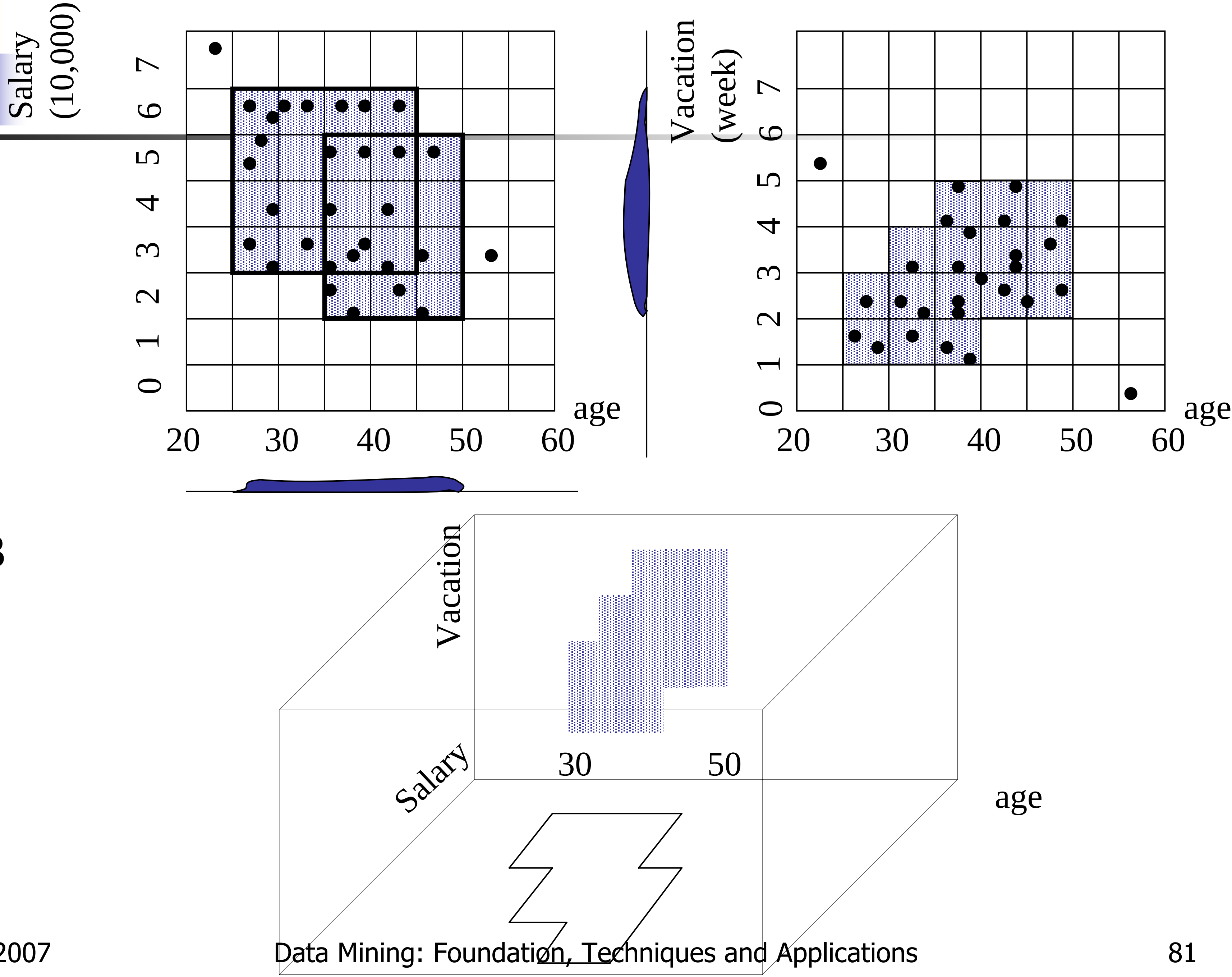
- How to apply wavelet transform to find clusters
  - Summaries the data by imposing a multidimensional grid structure onto data space
  - These multidimensional spatial data objects are represented in a n-dimensional feature space
  - Apply wavelet transform on feature space to find the dense regions in the feature space
  - Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse

# 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*

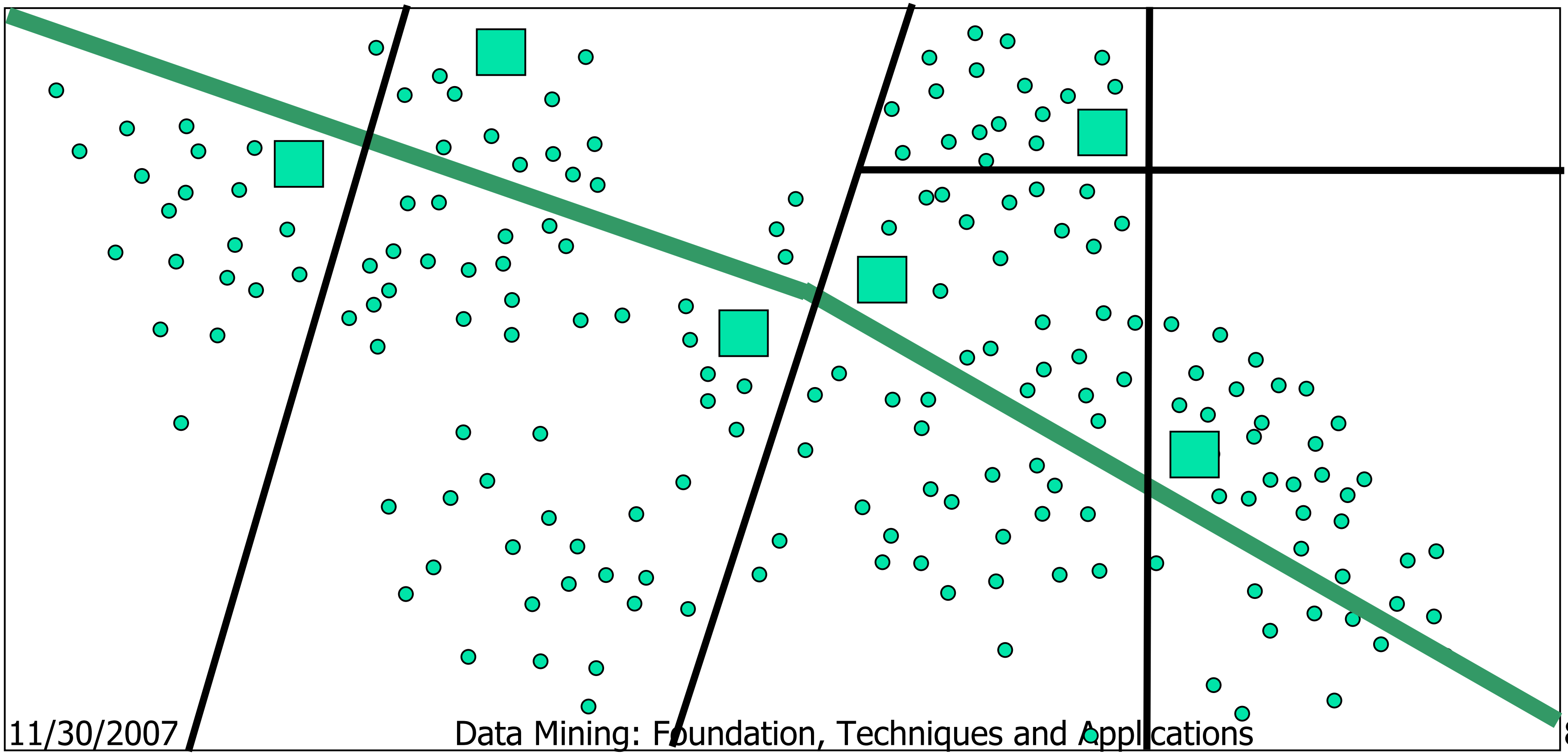
- Strength
  - It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - It is *insensitive* to the order of records in input and does not presume some canonical data distribution
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- Summary

# Why Constraint-Based Cluster Analysis?

- Need user feedback: Users know their applications the best
- Less parameters but more user-desired constraints, e.g., an ATM allocation problem: obstacle & desired clusters

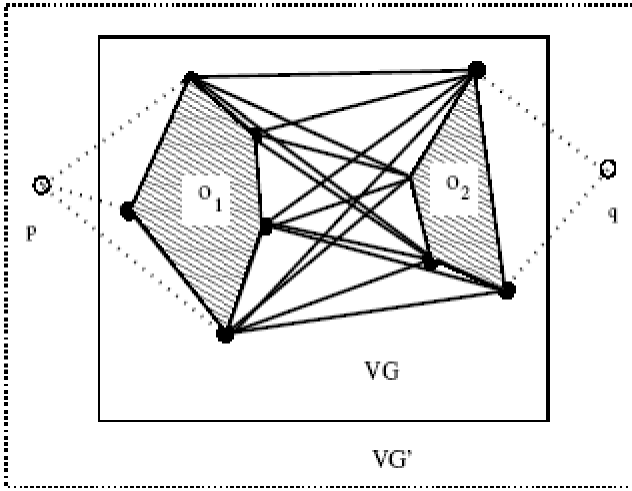


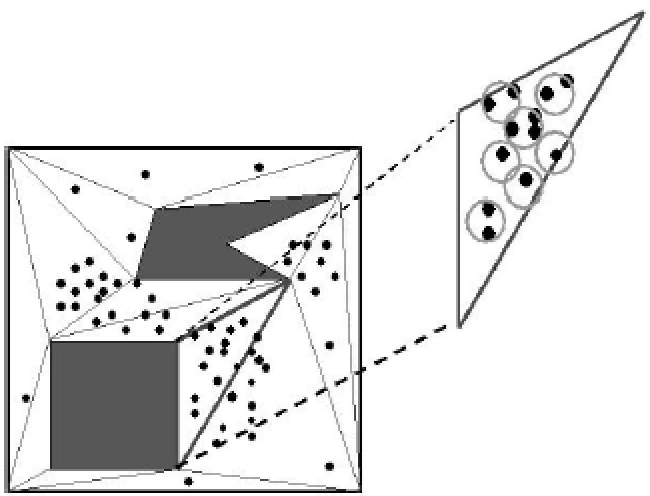
# A Classification of Constraints in Cluster Analysis

- Clustering in applications: desirable to have user-guided (i.e., constrained) cluster analysis
- Different constraints in cluster analysis:
  - Constraints on individual objects (do selection first)
    - Cluster on houses worth over $300K
  - Constraints on distance or similarity functions
    - Weighted functions, obstacles (e.g., rivers, lakes)
  - Constraints on the selection of clustering parameters
    - # of clusters, MinPts, etc.
  - User-specified constraints
    - Contain at least 500 valued customers and 5000 ordinary ones
  - Semi-supervised: giving small training sets as “constraints” or hints

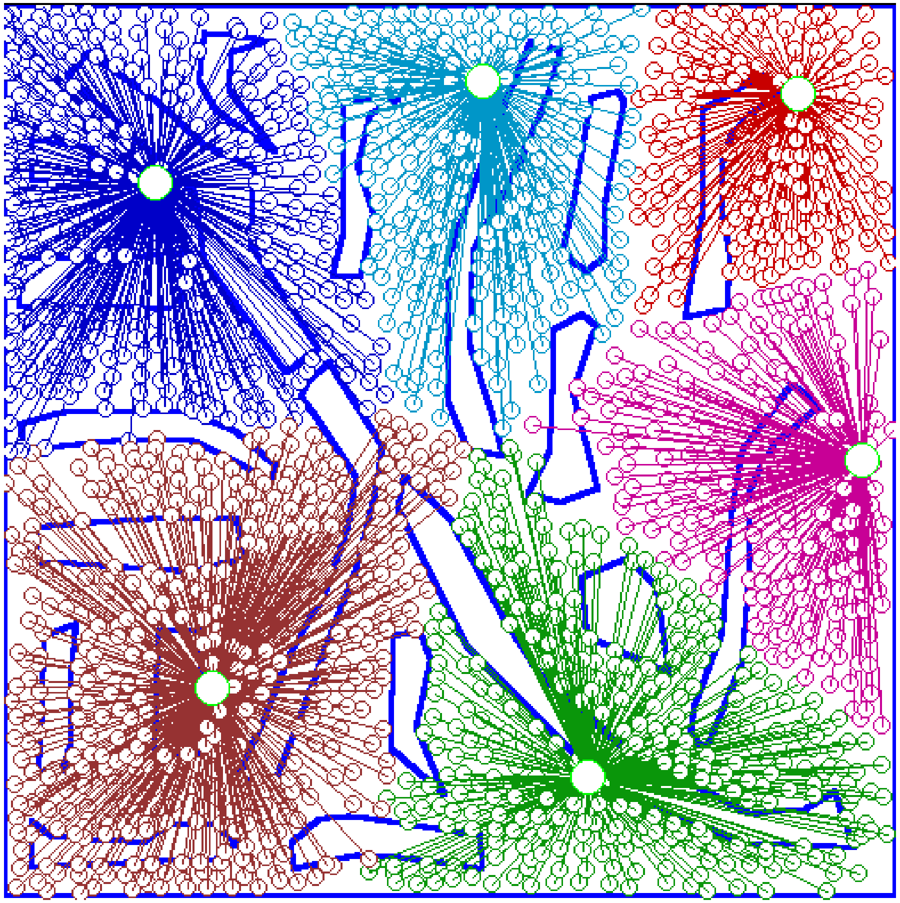
# Clustering With Obstacle Objects

- k-medoids is more preferable since k-means may locate the ATM center in the middle of a lake
  - Anthony K. H. Tung, Jean Hou, Jiawei Han, "Clustering in the Presence of Obstacles". In Proc. of 17th International Conference on Data Engineering (ICDE'01, Heidelberg, Germany p359-367.
- Visibility graph and shortest path
- Triangulation and micro-clustering
- Two kinds of join indices (shortest-paths) worth pre-computation
  - VV index: indices for any pair of obstacle vertices
  - MV index: indices for any pair of micro-cluster and obstacle indices

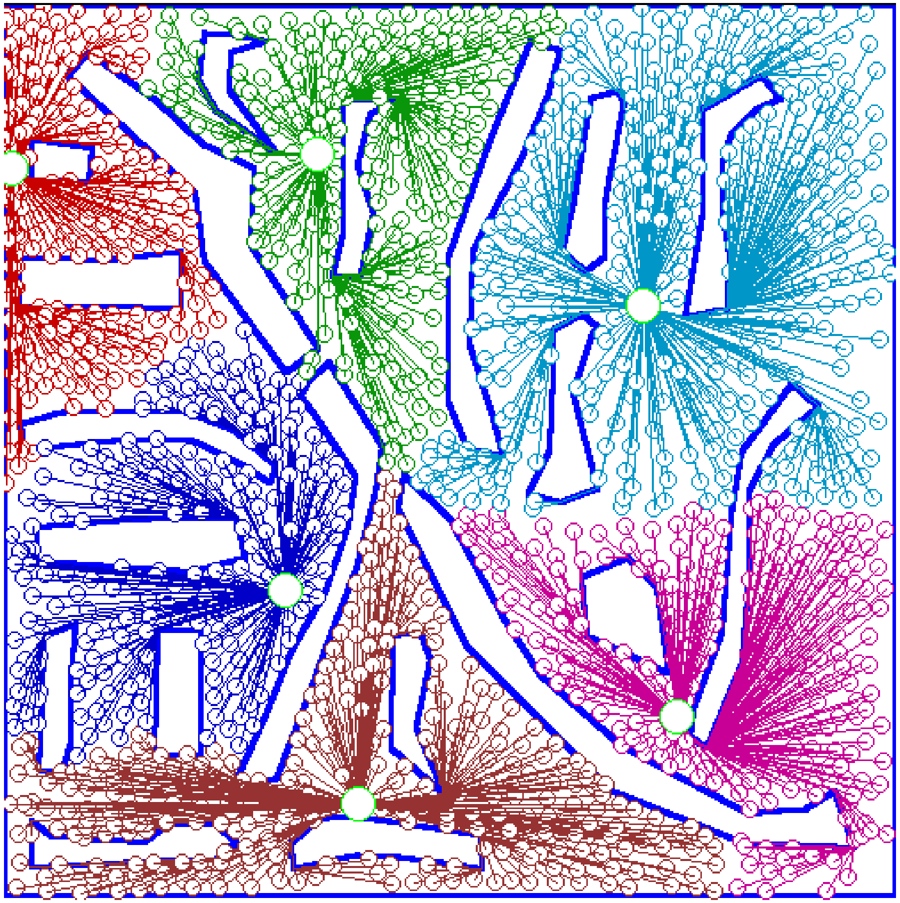


# An Example: Clustering With Obstacle Objects



***Not*** Taking obstacles into account



Taking obstacles into account

# Clustering with User-Specified Constraints

- Example: Locating k delivery centers, each serving at least m valued customers and n ordinary ones
  - Anthony K. H. Tung, Raymond T. Ng, Laks V. S. Lakshmanan,, Jiawei Han, "Constrained Clustering on Large Database" , Proc. 8th Intl. Conf. on Database Theory (ICDT'01), London, UK, Jan. 2001, p405-419.
- Proposed approach
  - Find an initial "solution" by partitioning the data set into k groups and satisfying user-constraints
  - Iteratively refine the solution by micro-clustering relocation (e.g., moving δ μ-clusters from cluster $C_i$ to $C_j$) and "deadlock" handling (break the microclusters when necessary)
  - Efficiency is improved by micro-clustering
- How to handle more complicated constraints?
  - E.g., having approximately same number of valued customers in each cluster?! — Can you solve it?

# Extensions

- k-anonymity: a concept in privacy preserving data publishing that require data records to be cluster into groups of at least size k. Can you see it as a clustering problem with constraints?
- Current methods for summarizing frequent patterns require you to find the patterns and then cluster them. Can it be done using ItCompress with constraints?
  - X. Yan, H. Cheng, J. Han, and D. Xin, "Summarizing Itemset Patterns: A Profile-Based Approach", in Proc. 2005 Int. Conf. on Knowledge Discovery and Data Mining (KDD'05), Chicago, IL, Aug. 2005. **(Best Student Paper Runner-Up Award)**
  - H. V. Jagadish, Raymond T. Ng, Beng Chin Ooi, Anthony K. H. Tung, "ItCompress: An Iterative Semantic Compression Algorithm". International Conference on Data Engineering (ICDE'2004), Boston, 2004

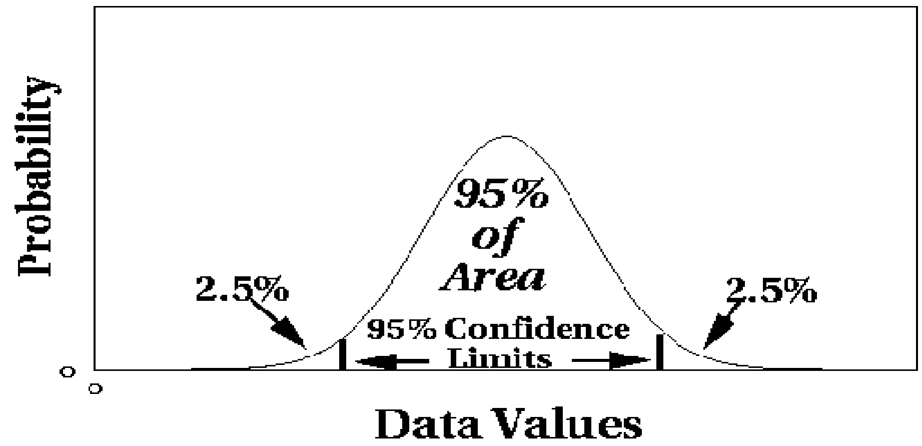
# Outline

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Constrained Clustering
- Outlier Analysis
- 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
  - Find top n outlier points
- 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 single attribute
  - In many cases, data distribution may not be known

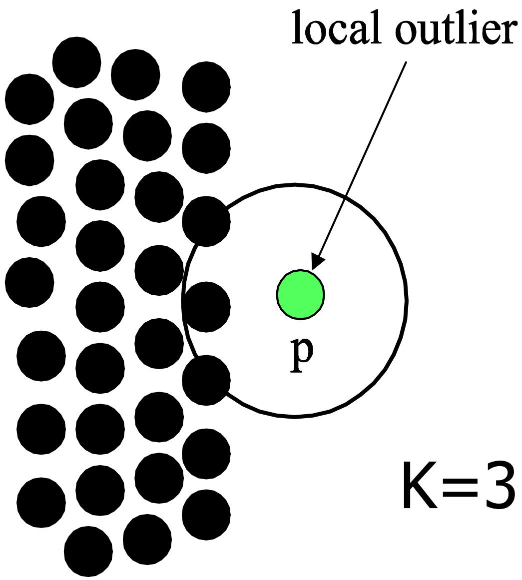
# Outlier Discovery: Distance-Based Approach

- 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

# Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that "deviate" from this description are considered outliers
- sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data

# Local Outlier Factor(LOF)




- Outliers are computed w.r.t to the densities of the neighborhood
- First proposed
  - Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander: LOF: Identifying Density-Based Local Outliers. SIGMOD Conference 2000: 93-104
- Extended to find top-n
  - Wen Jin, Anthony K. H. Tung , Jiawei. Han, "Finding Top-n Local Outliers in Large Database", in 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (SIGKDD'01)

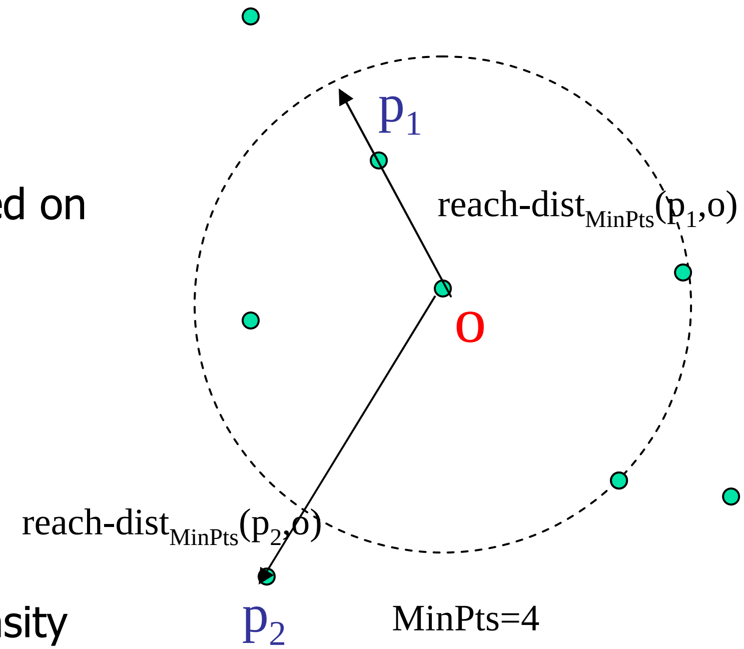
# Local View of Outliers

- Outliers are computed respect to the densities of the neighborhoods
- Reachability distance of p w.r.t o
  - Reach-dist$_k$ =max (k-distance(o), d(p,o))
- (lrd) Local Reachability Density of p
  - the inverse of the average reachability distance based on the MinPts-nearest neighbors of p

$$lrd_{MinPts}(p) = 1 / \left( \frac{\sum_{o \in N_{MinPts}(p)} reach\text{-}dist_{MinPts}(p, o)}{|N_{MinPts}(p)|} \right)$$

- (LOF) Local Outlier Factor of p
  - the average of the ratio of the local reachability density of p and those of p's MinPts-nearest neighbors

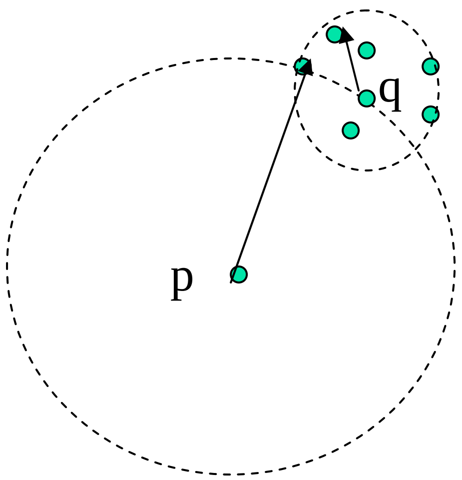
$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}$$



# The properties of LOF

- The **lower** p's local reachability density is, the higher the LOF value of p is. ( That is, the **higher** p's MinPts reachability distance is, the higher the LOF.)
- The **higher** q's local reachability density is, the higher the LOF. (That is, the **lower** q's MinPts-nearest reachability distance is, the higher the LOF .)

*q is p's MinPts-nearest neighbor*

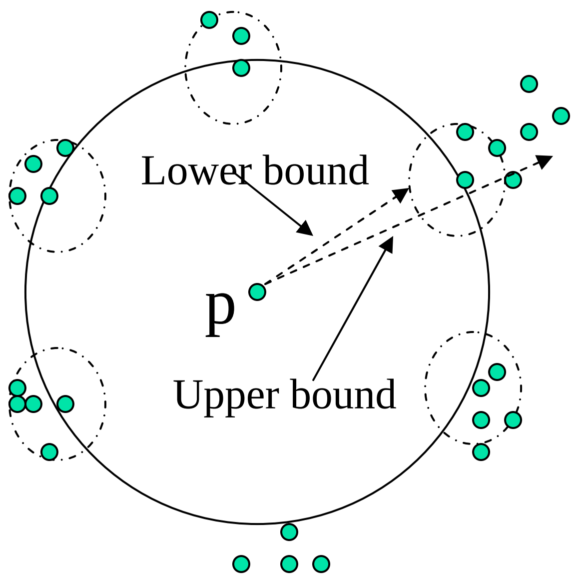


# Finding Top-n Outlier based on LOF

- The original algorithm compute LOF for all points. If we are only interested in top n LOF, n<<DB, the reachability distance computations for most of the remaining points which do not affect those Top-n LOF  calculation, are of little use and can be altogether avoided.
- Try to partitioning data space into “micro-clusters” so that the lower/upper bound of reachability distance of each micro-cluster instead of each data is determined instead of the huge cost of  computation data by data .
- Prune out a significant number of micro-clusters whose LOF are so small that cannot possibly become TOP-n LOF.

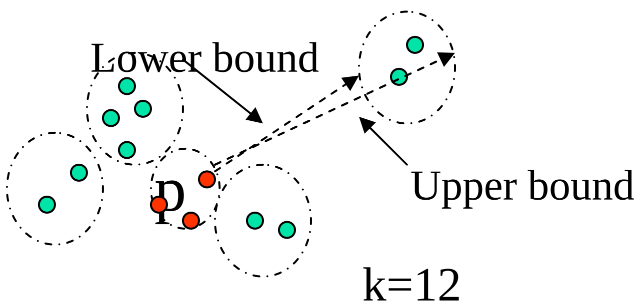
# What is a candidate LOF?

- Those data with high reachability distance have high probability to become Top-n LOF. But those data whose MinPts reachability distance are neither very low nor very high need to be paid much attention.
- The lower and upper bound of reachability distance is required to further identify candidate LOF. In term, this mean that we need to know the upper and lower bound for k-distance(p).



# How to determine lower/upper bound for k-distance of p?

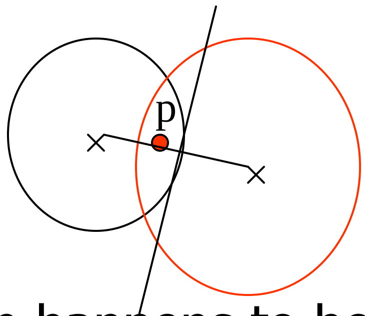
- The lower bound is the minimal distance between p and the furthest micro-cluster that contain k points around p.




- The upper bound of  k-distance of  p is any distance that is guaranteed to contain k points.
- The k-distance of each data is approximated by comparing with other micro-clusters instead of  computing data pair by pair
- Note, if p can find k points within a micro-cluster p belongs to, the lower/upper bound is the distance between nearest neighbor and p, and the micro-cluster's inter/external  respectively

# What if overlapping micro-clusters occur?

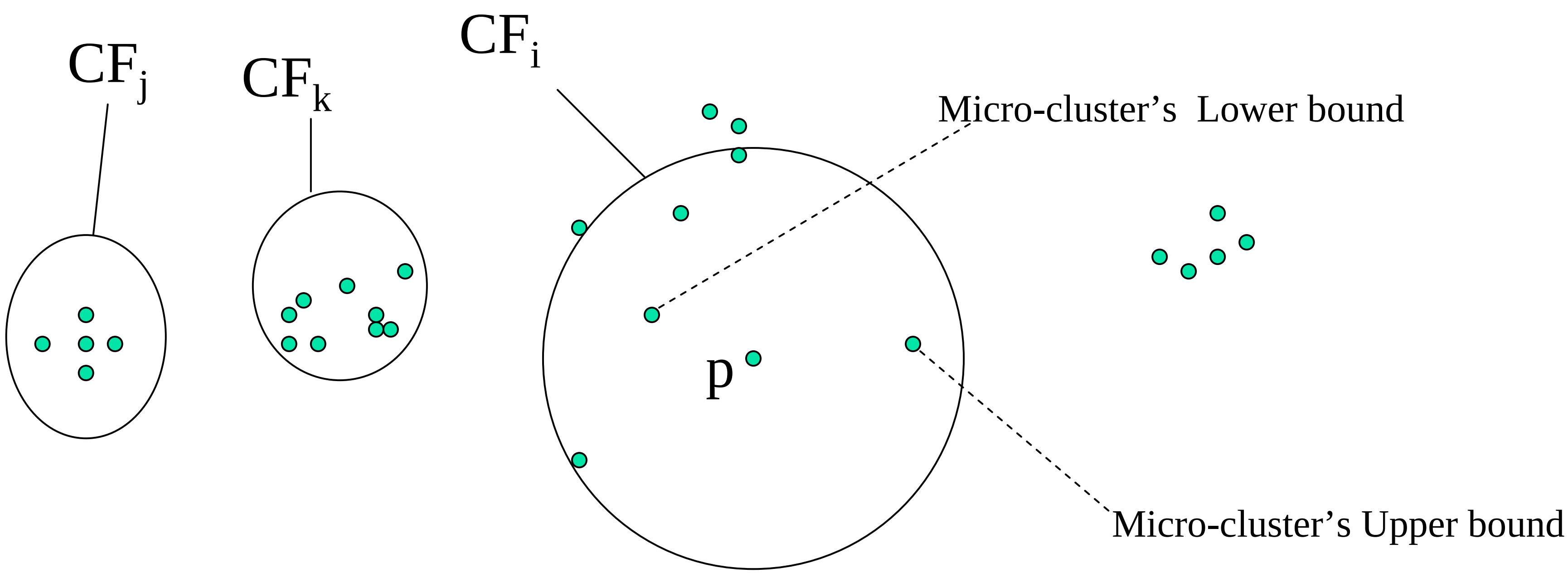
- Based on the mean of two centers of micro-clusters, a hyper-plane is created in constant time, the distance between p and the hyper-plane is taken as the lower bound of reachability distance.




- The rare but worst case is if p happens to be on the plane, then to calculate the distance between p and the data in its nearest micro-cluster, choose the minimal one as the lower bound

# How to determine lower/upper bound for a micro-cluster's k-distance?

- In a micro-cluster, compare each point's lower/upper bound of k- distance within Minpts range, select the minimal/maximal one as the micro-cluster's lower/upper bound.



# How to determine lower/upper bound for a micro-cluster' LOF?

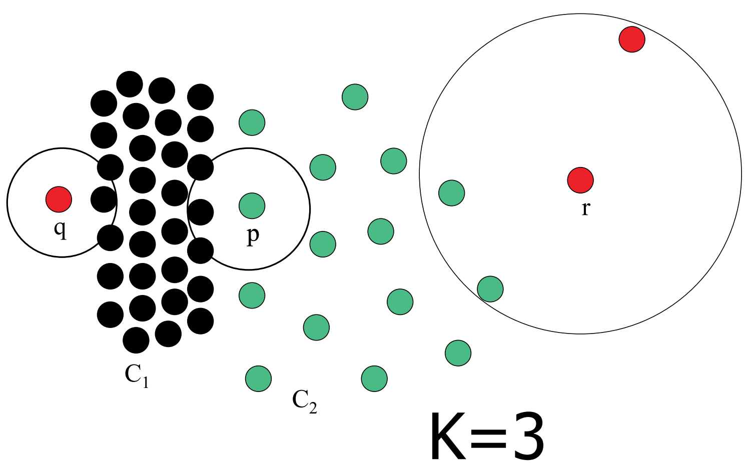
**Theorem** : if $p \in$ DB, MC is micro-cluster and $p \in MC$ , *then*

$$\frac{k-\text{distance}(MC).lower}{k-\text{distance}(MC).upper} \prec LOF(p) \prec \frac{k-\text{distance}(MC).upper}{k-\text{distance}(MC).lower}$$

- Based on this property, LOF bound for each micro-cluster can be made

# Ranking Outliers Using Symmetric Neighborhood Relationship

- Take into account both nearest neighbor and reverse nearest neighbor distance
  - Wen Jin, Anthony K. H. Tung, Jiawei Han, and Wei Wang, "Ranking Outliers Using Symmetric Neighborhood Relationship," in Proc. 2006 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'06), Singapore, April 2006.




- **Case 1**: if the densities of the nearest neighboring objects for both p and q are the same, but q is slightly closer to cluster C1 than p
- **Case 2**: the density of r is lower than p, the average density of its neighboring objects (consisting of 2 objects from C2 and an outlier) is less than those of p. Thus, when the LOF measure is computed, p has stronger outlierness than r. But again it is wrong!

# Influential Measure of Outlierness by Symmetric Relationship

- The density of p, denoted as den(p), is the inverse of the k-distance of p, i.e., $den(p) = 1/k_{dist}(p)$.
- k-influence space for p, denoted as $IS_k(p)$, consists of $NN_k(p)$ and $RNN_k(p)$.
- The influenced outlierness (***INFLO***) is defined as:

$$INFLO_k(p) = \frac{den_{avg}(IS_k(p))}{den(p)} \quad \text{where} \quad den_{avg}(IS_k(p)) = \frac{\sum_{o \in IS_k(p)} den(o)}{|IS_k(p)|}$$

- The higher INFLO is, the more likely that this object is an outlier. The lower INFLO is, the more likely that this object is a member of a cluster. Specifically, INFLO $\approx$ 1 means the object locates in the core part of a cluster.

# Mining Algorithms for Top-n INFLO

- Naïve index-based method
- Two-way search method
- Micro-cluster method.

# 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
- There are still lots of research issues on cluster analysis, such as constraint-based clustering
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches

# Essential Reading

- [HK01]: "Data Mining: Concepts and Techniques", Chapter 7.1- 7.11
- Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander: OPTICS: Ordering Points To Identify the Clustering Structure. SIGMOD Conference 1999: 49-60
- Anthony K. H. Tung, Jean Hou, Jiawei Han, "Clustering in the Presence of Obstacles". In Proc. of 17th International Conference on Data Engineering (ICDE'01, Heidelberg, Germany p359-367
- Anthony K. H. Tung, Raymond T. Ng, Laks V. S. Lakshmanan,, Jiawei Han, "Constrained Clustering on Large Database" , Proc. 8th Intl. Conf. on Database Theory (ICDT'01), London, UK, Jan. 2001, p405-419.
- Wen Jin, Anthony K. H. Tung, Jiawei Han, and Wei Wang, "Ranking Outliers Using Symmetric Neighborhood Relationship," in Proc. 2006 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'06), Singapore, April 2006.

# References


- Paul S. Bradley, Usama M. Fayyad, Cory Reina: Scaling Clustering Algorithms to Large Databases. KDD 1998: 9-15
- Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander: LOF: Identifying Density-Based Local Outliers. SIGMOD Conference 2000: 93-104
- Wen Jin, Anthony K. H. Tung , Jiawei. Han, "Finding Top-n Local Outliers in Large Database", in 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (SIGKDD'01)
- Zhenjie Zhang, Bing Tian Dai and Anthony K.H. Tung. "On the Lower Bound of Lower Optimums in K-Means Algorithm". In ICDM 2006. [Codes][PPT]