Data Mining:

Lecture 6-8: CLUSTER ANALYSIS —

Ph.D. Shatovskaya T.

Department of Computer Science

Chapter 8. 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
- Model-Based Clustering Methods
- 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

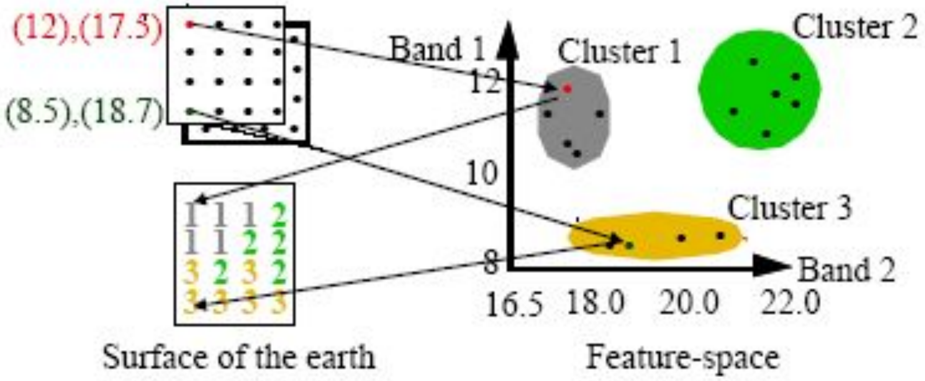
- Pattern Recognition
- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to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

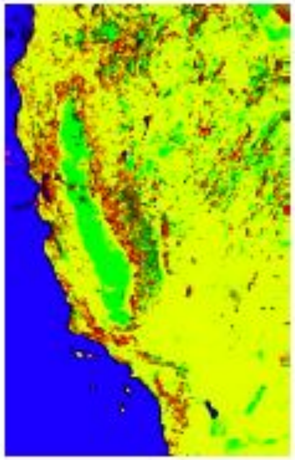
Examples of Clustering Applications

- 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

A Typical Application: Thematic Maps

- Satellite images of a region in different wavelengths
  - Each point on the surface maps to a high-dimensional feature vector $p = (x_1, \ldots; x_d)$ where x_i is the recorded intensity at the surface point in band i.
  - Assumption: each different land-use reflects and emits light of different wavelengths in a characteristic way.

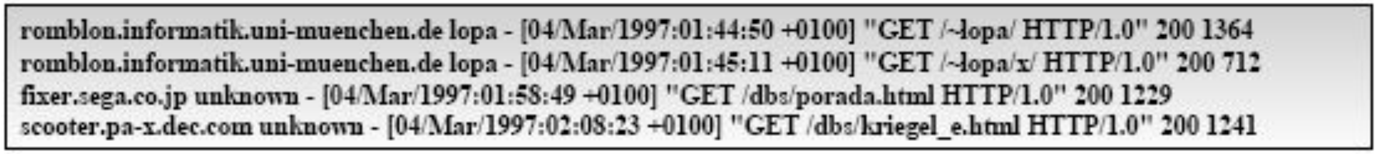




Application: Web Usage Mining

Determine Web User Groups

Sample content of a web log file



```
romblon.informatik.uni-muenchen.de lopa - [04/Mar/1997:01:44:50 +0100] "GET /~lopa/ HTTP/1.0" 200 1364
romblon.informatik.uni-muenchen.de lopa - [04/Mar/1997:01:45:11 +0100] "GET /~lopa/x/ HTTP/1.0" 200 712
fixer.sega.co.jp unknown - [04/Mar/1997:01:58:49 +0100] "GET /dbs/porada.html HTTP/1.0" 200 1229
scooter.pa-x.dec.com unknown - [04/Mar/1997:02:08:23 +0100] "GET /dbs/kriegel_e.html HTTP/1.0" 200 1241
```

Generation of sessions

⇨ Session::= <IP_address, user_id, [URL_1, . . ., URL_k]>

which entries form a single session?

Distance function for sessions: $d(x,y) = \dfrac{|x \cup y| - |x \cap y|}{|x \cup y|}$

Major Clustering Approaches

- Partitioning algorithms
  - Find *k* partitions, minimizing some objective function
- Hierarchy algorithms
  - Create a hierarchical decomposition of the set of objects
- Density-based
  - Find clusters based on connectivity and density functions
- Other methods
  - Grid-based
  - Neural networks (SOM's)
  - Graph-theoretical methods
  - . . .

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.

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

Chapter 8. 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
- Model-Based Clustering Methods
- 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.

Example distance functions I

- For standardized numerical attributes, i.e., vectors $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$ from a d-dimensional vector space:
  - General L_p-Metric (Minkowski-Distance): $dist(x,y) = \sqrt[p]{\sum_{i=1}^{d} (x_i - y_i)^p}$
  - Euclidean Distance ($p = 2$): $dist(x,y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$
  - Manhattan-Distance ($p = 1$): $dist(x,y) = \sum_{i=1}^{d} \left| x_i - y_i \right|$
  - Maximum-Metric ($p = \infty$): $dist(x,y) = \max\left\{ \left| x_i - y_i \right|, 1 \le i \le d \right\}$
- For sets *x* and *y:* $dist(x,y) = \frac{\left| x \cup y \right| - \left| x \cap y \right|}{\left| x \cup y \right|}$

Example distance functions II

- For categorical attributes:

$$dist(x,y)=\sum_{i=1}^{d}\delta(x_i,y_i) \text{ where } \delta(x_i,y_i)=\begin{cases}0 & \text{if } x_i=y_i\\ 1 & \text{else}\end{cases}$$

- For text documents:
  - A document D is represented by a vector $r(D)$ of frequencies of the terms occuring in D, e.g.,

    $$r(d)=\{\log(f(t_i,D)),\ t_i\in T\}$$

    where $f(t_i, D)$ is the frequency of term t_i in document D
  - The distance between two documents D_1 and D_2 is defined by the cosine of the angle between the two vectors $x = r(D_1)$ and $y = r(D_2)$:

    $$dist(x,y)=1-\frac{\langle x,y\rangle}{|x|\cdot|y|}$$

    where $\langle\ .\ ,\ .\ \rangle$ is the inner product and $|\ .\ |$ is the length of vectors

Measuring Similarity

- To measure similarity, often a distance function *dist* is used
  - Measures "dissimilarity" between pairs objects *x* and *y*
    - Small distance $dist(x, y)$: objects x and y are more similar
    - Large distance $dist(x, y)$: objects x and y are less similar
- Properties of a distance function
  - $dist(x, y) \geq 0$
  - $dist(x, y) = 0$ iff $x = y$ (definite) (iff = if and only if)
  - $dist(x, y) = dist(y, x)$ (symmetry)
  - If *dist* is a metric, which is often the case:
    $dist(x, z) \leq dist(x, y) + dist(y, z)$ (triangle inequality)
- Definition of a distance function is highly application dependent
  - May require standardization/normalization of attributes
  - Different definitions for interval-scaled, boolean, categorical, ordinal and ratio variables

Type of data in clustering analysis

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

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

Binary Variables

- A contingency table for binary data

		Object j		
		1	0	sum
Object i	1	a	b	$a+b$
	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*):

$$d(i,j) = \frac{b+c}{a+b+c}$$

Binary Variables

- Rassel and Rao coefficient: J(i,j)= a/ a+b+c+d

- Bravais coefficient: C(i,j)= ad-bc/ $\sqrt{(a+b)(a+c)(d+b)(d+c)}$

- Association coefficient Yule: Q(i,j)= ad-bc/ ad+bc

- Hemming distance: H(i,j)= a+d

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
  - replace 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?—the scale can be distorted)
  - 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

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

  - *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

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

Chapter 8. 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
- Model-Based Clustering Methods
- 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
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Chapter 8. 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
- Model-Based Clustering Methods
- 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

K-Means Clustering: Basic Notions

- Objects $p=(x^p{}_1, \ldots, x^p{}_d)$ are points in a d-dimensional vector space (the mean of a set of points must be defined)
- *Centroid* μ_C: Mean of all points in a cluster C
- Measure for the compactness of a **cluster** C:

$$TD^2(C) = \sum_{p \in C} dist(p, \mu_C)^2$$

- Measure for the compactness of a **clustering**

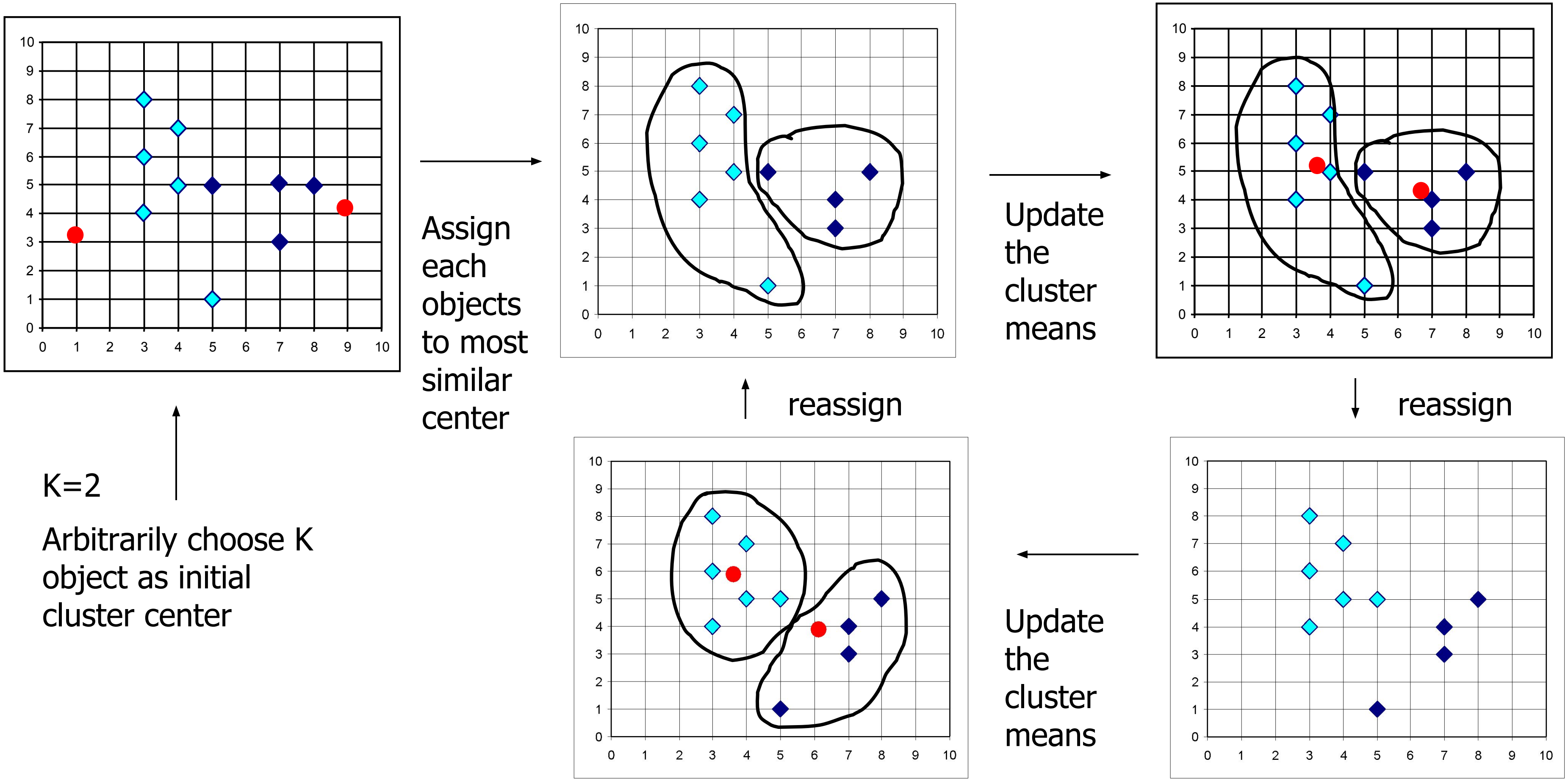
$$TD^2 = \sum_{i=1}^{k} TD^2(C_i)$$

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



K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

reassign

reassign

Update the cluster means

Comments on the *K-Means* Method

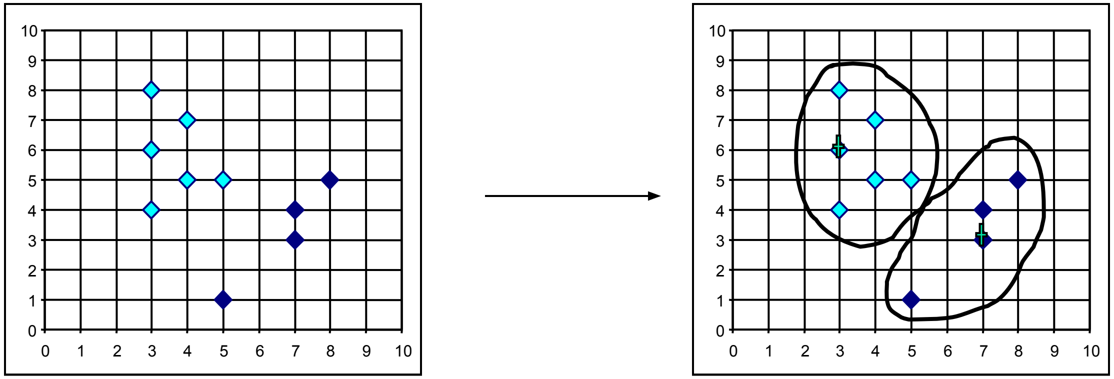
- Strength: *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
    - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
    - Applicable only when *mean* is defined, then what about categorical data?
    - 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*

Variations of the *K-Means* Method

- A few 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* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

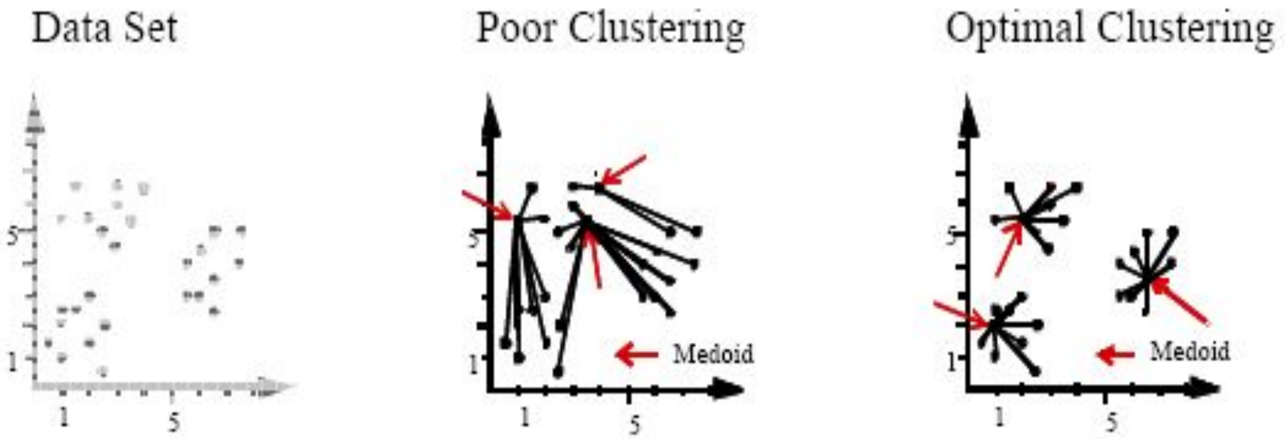
What is the problem of k-Means Method?

- 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.



K-Medoid Clustering: Basic Idea

- Objective: For a given *k*, find *k* representatives in the dataset so that, when assigning each object to the closest representative, the sum of the distances between representatives and objects which are assigned to them is minimal.



Data Set

Poor Clustering

Optimal Clustering

K-Medoid Clustering: Basic Notions

- Requires arbitrary objects and a distance function
- *Medoid* m_C: representative object in a cluster *C*
- Measure for the compactness of a Cluster C:

$$TD(C) = \sum_{p \in C} dist(p, m_C)$$

- Measure for the compactness of a clustering

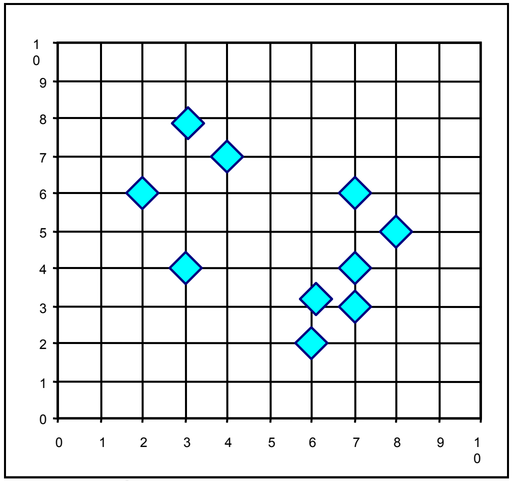
$$TD = \sum_{i=1}^{k} TD(C_i)$$

K-Medoid Clustering: PAM Algorithm

- [Kaufman and Rousseeuw, 1990]
- Given *k*, the *k*-medoid algorithm is implemented in 5 steps:
  1. Select *k* objects arbitrarily as medoids (representatives); assign each remaining (non-medoid) object to the cluster with the nearest representative, and compute $TD_{current}$.
  2. For each pair (medoid *M*, non-medoid *N*)
     - compute the value $TD_{N\leftrightarrow M}$, i.e., the value of TD for the partition that results when "swapping" *M* with *N*
  3. Select the non-medoid *N* for which $TD_{N\leftrightarrow M}$ is minimal
  4. If $TD_{N\leftrightarrow M}$ is smaller than $TD_{current}$
     - Swap *N* with *M*
     - Set $TD_{current} := TD_{N\leftrightarrow M}$
     - Go back to Step 2
  5. Stop.

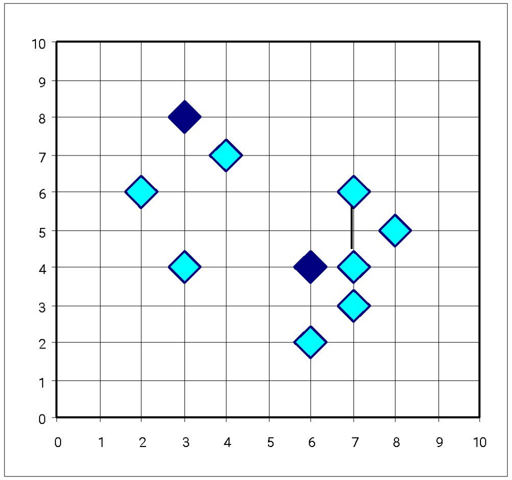
Typical k-medoids algorithm (PAM)

Total Cost = 20

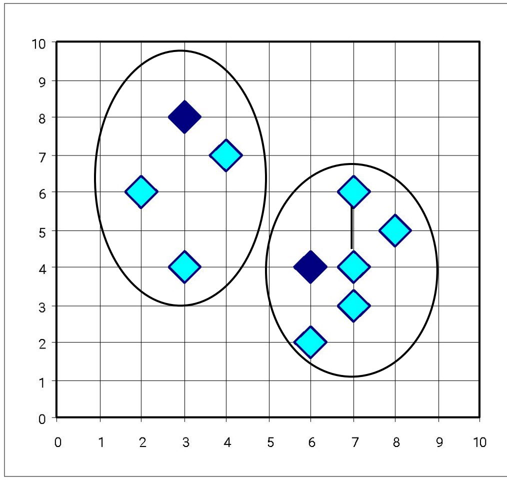


K=2

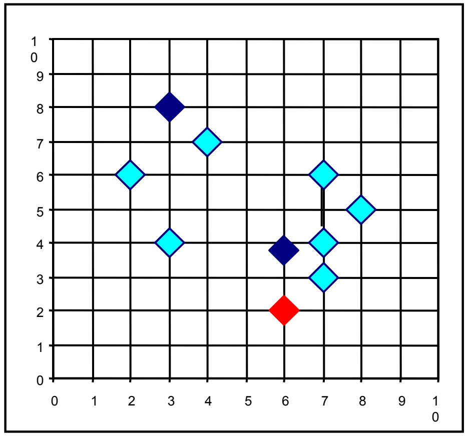
Arbitrary choose k object as initial medoids



Assign each remaining object to nearest medoids

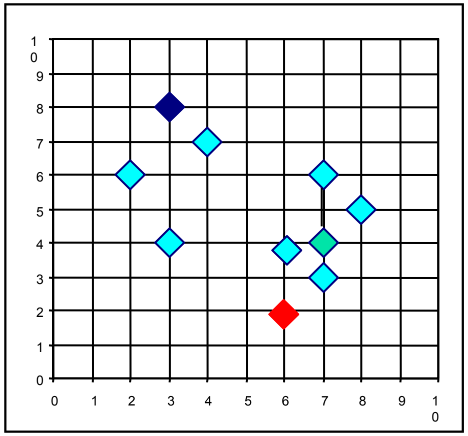


Randomly select a nonmedoid object,O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

What is the problem with PAM?

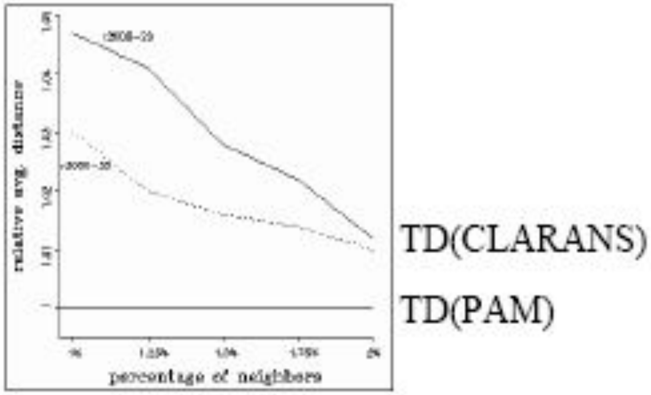
- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

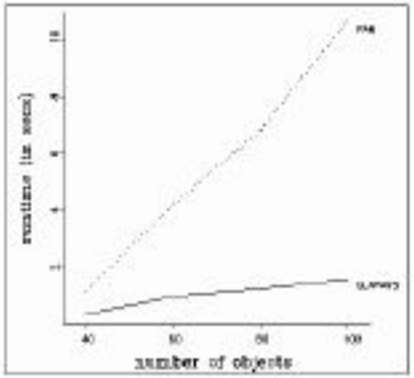
    where n is # of data,k is # of clusters
- Sampling based method,

  CLARA(Clustering LARge Applications)

K-Medoid Clustering: Discussion

- Strength
  - Applicable to arbitrary objects + distance function
  - Not so sensitive to noisy data and outliers as *k*-means
- Weakness
  - Inefficient
  - Like k-means: need to specify the number of clusters *k* in advance, and clusters are forced to have convex shapes
  - Result and runtime for CLARA and CLARANS may vary largely due to the randomization





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

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)

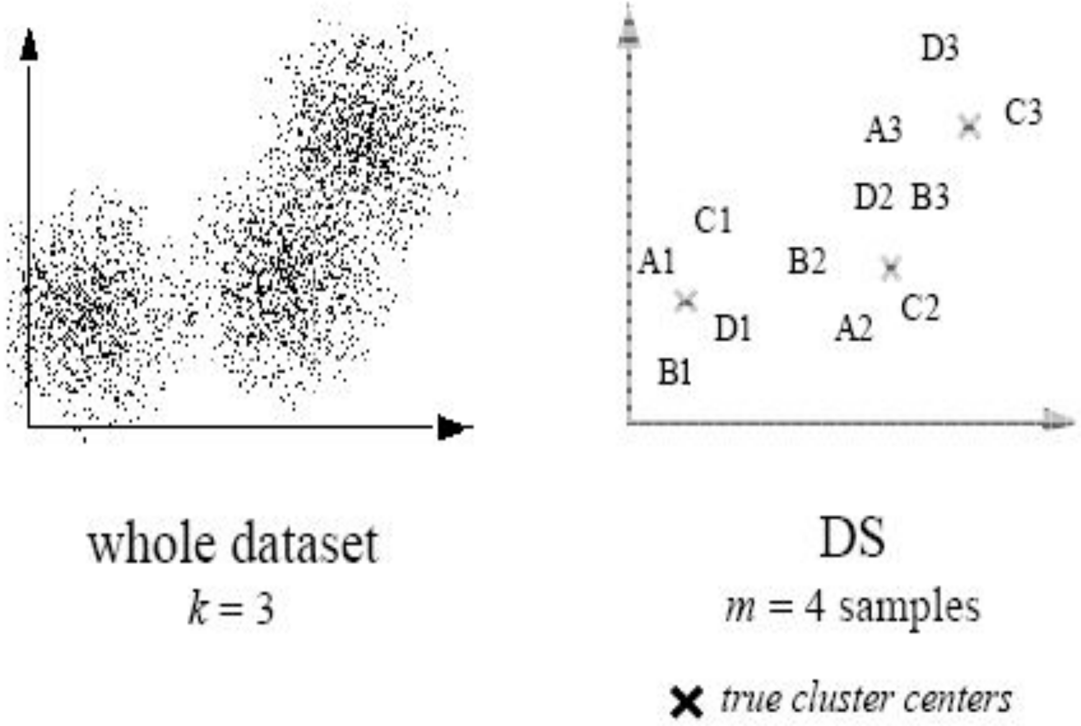
Initialization of Partitioning Clustering Methods

- [Fayyad, Reina and Bradley 1998]
  - Draw *m* different (small) samples of the dataset
  - Cluster each sample to get *m* estimates for *k* representatives

    $A = (A_1, A_2, . . ., A_k), B = (B_1, . . ., B_k), ..., M = (M_1, . . ., M_k)$
  - Then, cluster the set $DS = A \cup B \cup ... \cup M$ *m* times, using the sets *A*, *B*, ..., *M* as respective initial partitioning
  - Use the best of these *m* clusterings as initialization for the partitioning clustering of the whole dataset

Initialization of Partitioning Clustering Methods

Example


whole dataset

$k = 3$

DS

$m = 4$ samples

Choice of the Parameter *k*

- Idea for a method:
  - Determine a clustering for each $k = 2, \ldots n\text{-}1$
  - Choose the „best" clustering
- But how can we measure the quality of a clustering?
  - A measure has to be independent of *k*.
  - The measures for the compactness of a clustering TD^2 and TD are monotonously decreasing with incresing value of *k*.
- Silhouette-Coefficient [Kaufman & Rousseeuw 1990]
  - Measure for the quality of a *k*-means or a *k*-medoid clustering that is independent of *k*.

The Silhouette Coefficient

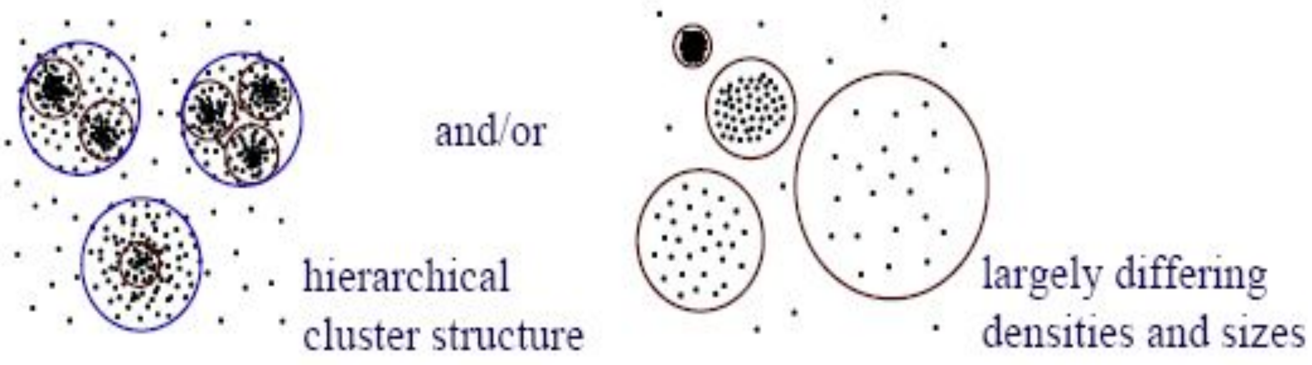
- *a(o):* average distance between object *o* and the objects in its cluster *A*
- *b(o):* average distance between object *o* and the objects in its "second closest" cluster *B*
- The silhouette of *o* is then defined as $s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$
  - measures how good the assignment of *o* to its cluster is
    - $s(o) = -1$: bad, on average closer to members of *B*
      $s(o) = 0$: in-between *A* and *B*
      $s(o) = 1$: good assignment of *o* to its cluster *A*
- Silhouette Coefficient s_C of a clustering: average silhouette of all objects
  - $0.7 < s_C \leq 1.0$ strong structure, $0.5 < s_C \leq 0.7$ medium structure
  - $0.25 < s_C \leq 0.5$ weak structure, $s_C \leq 0.25$ no structure

Chapter 8. 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
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

From Partitioning to Hierarchical Clustering

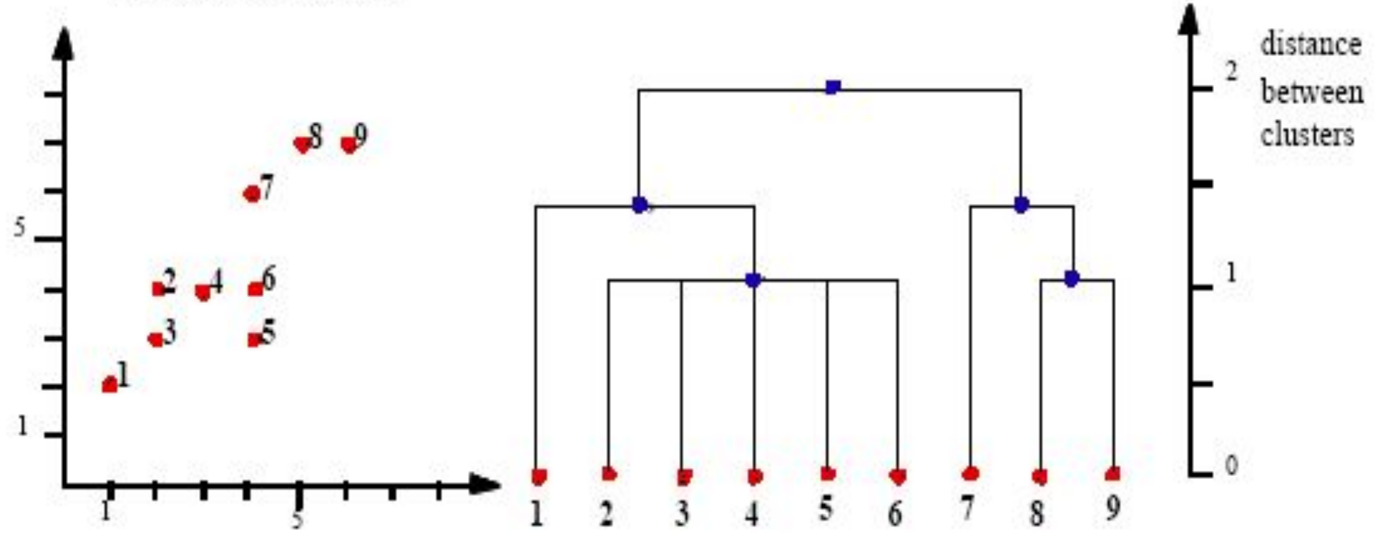
- Global parameters to separate all clusters with a partitioning clustering method may not exist


and/or
hierarchical cluster structure
largely differing densities and sizes

- Need a hierarchical clustering algorithm in these situations

Hierarchical Clustering: Example

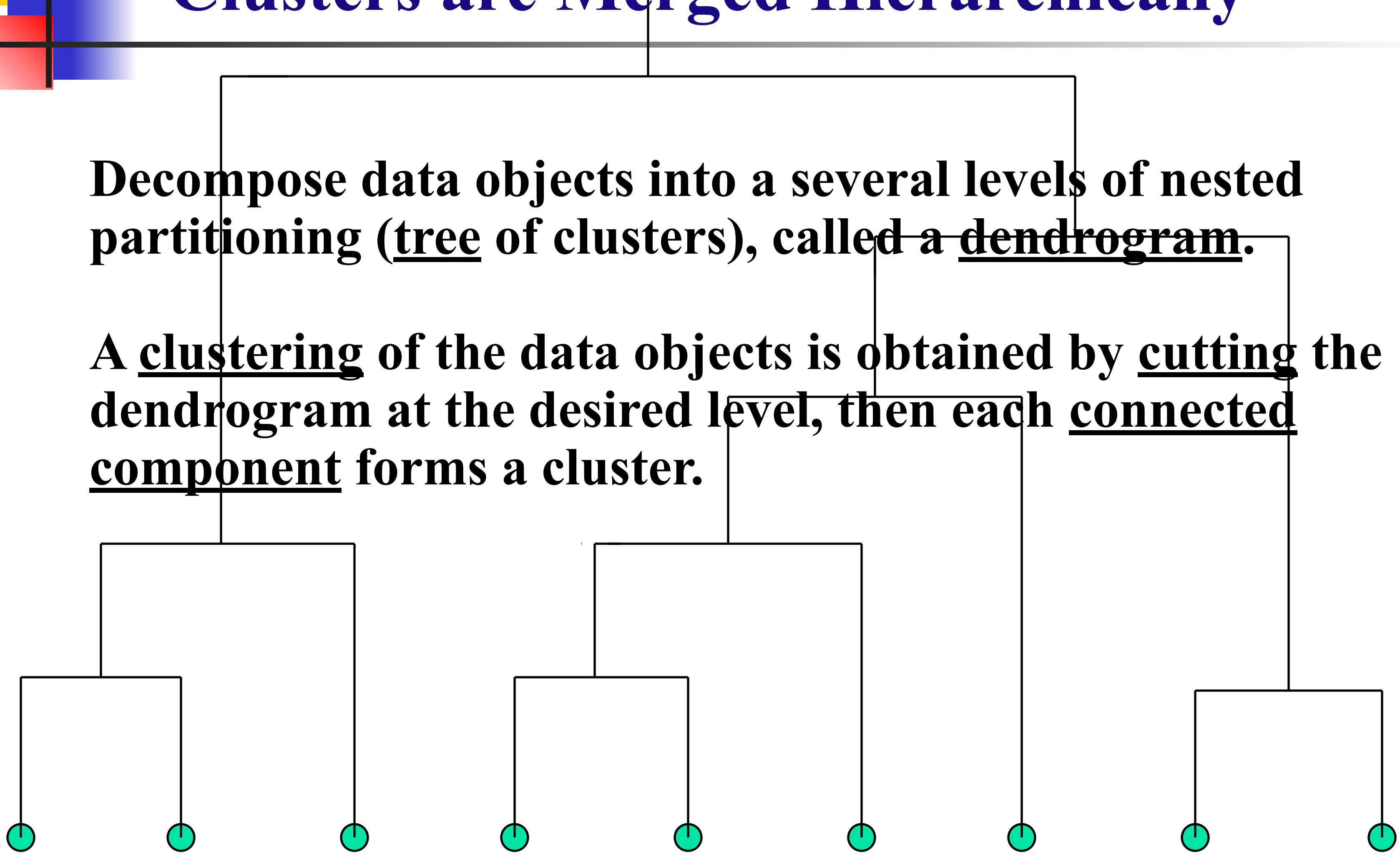
- Interpretation of the dendrogram
  - The root represents the whole data set
  - A leaf represents a single objects in the data set
  - An internal node represent the union of all objects in its sub-tree
  - The height of an internal node represents the distance between its two child nodes



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

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.



A Dendrogram Algorithm for Binary variables

1. To estimate similarity of objects on the basis of binary attributes and measures of similarity of objects such as Simple matching coefficient, Jaccard coefficient, Rassel and Rao coefficient, Bravais coefficient, association coefficient Yule, Hemming distance.

2.To make a incedence matrix for all objects, where it's elements is similarity coefficients.

3. Graphically represent a incedence matrix where on an axis x – number of objects, on an axis Y –the measures of similarity. Find in a matrix two most similar objects (with the minimal distance) and put them on the schedule. Iteratively continue construction of the schedule for all objects of the analysis

Example for binary variables

We have 3 objects with 16 attributes . Define the similarity of objects.

ecoli1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	1
ecoli2	0	1	0	1	1	0	0	1	0	0	0	0	0	0	1	0
ecoli3	1	1	0	1	1	0	0	1	0	0	0	0	0	0	1	1

1. Define the similarity on the base of Simple matching coefficient

ecoli1
ecoli2

	1	0
1	4	1
0	2	9

$J_{12}=13/16=0.81$

ecoli1
ecoli3

	1	0
1	4	2
0	1	8

$J_{13}=12/15=0.8$

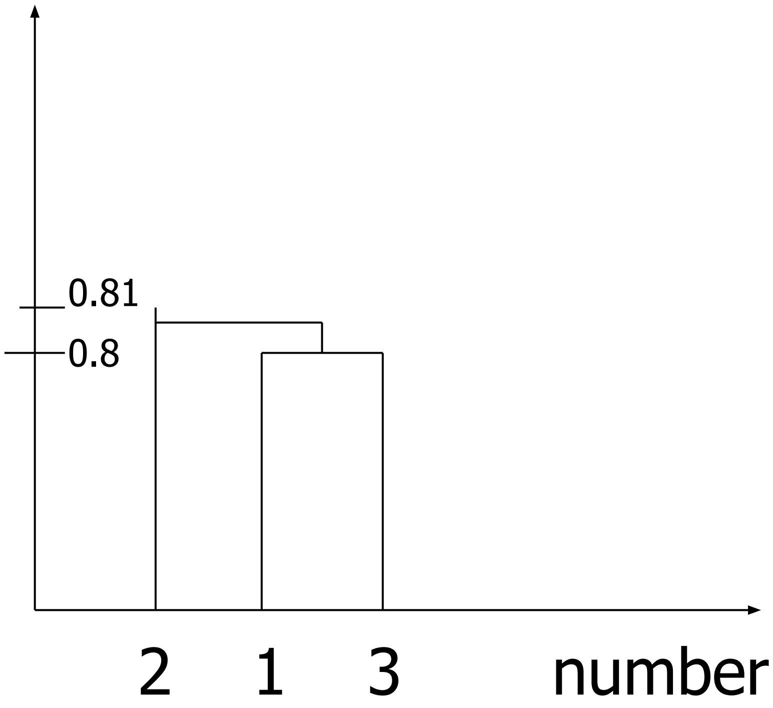
Example for binary variables

		1	0
ecoli2	1	5	2
ecoli3	0	0	9

$J_{23}=14/16=0.875$

2. Incedence matrix

	ecoli1	ecoli2	ecoli3
ecoli1	0	0.81	0.8
ecoli2		0	0.875
ecoli3			



A Dendrogram Algorithm for Numerical variables

1. To estimate similarity of objects on the basis of numerical attributes and measures of similarity of objects such as distances (slide 14).
2.To make a incedence matrix for all objects, where it's elements is distances.
3. Graphically represent a incedence matrix where on an axis x – number of objects, on an axis Y –the measures of similarity. Find in a matrix two most similar objects (with the minimal distance) and put them on the schedule. Iteratively continue construction of the schedule for all objects of the analysis

Single Link Method and Variants

- Given: a distance function *dist*(*p*, *q*) for database objects
- The following distance functions for clusters (i.e., sets of objects) *X* and *Y* are commonly used for hierarchical clustering:

Single-Link: $$dist\_sl(X,Y) = \min_{x \in X, y \in Y} dist(x,y)$$

Complete-Link: $$dist\_cl(X,Y) = \max_{x \in X, y \in Y} dist(x,y)$$

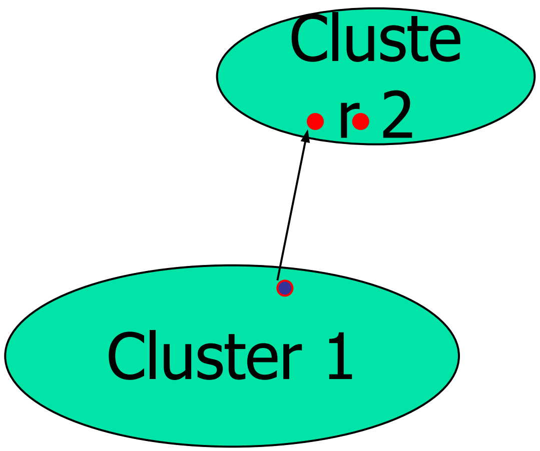
Average-Link: $$dist\_al(X,Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} dist(x,y)$$

A Dendrogram Algorithm for Numerical variables

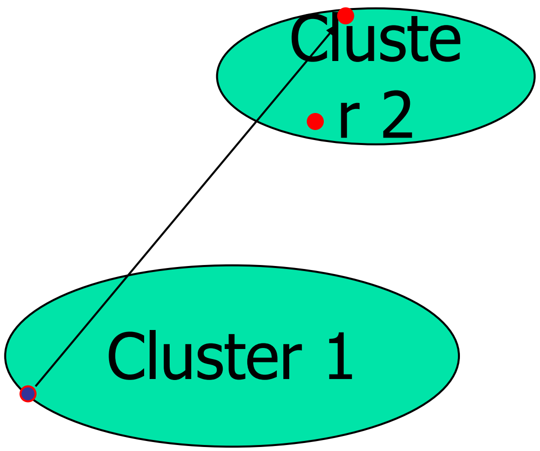
Let us consider five points {x_1,….,x_5} with the attributes

X1=(0,2), x2=(0,0) x3=(1.5,0) x4=(5,0) x5=(5,2)

Using Euclidian measure



a) single-link distance



b) complete-link distance

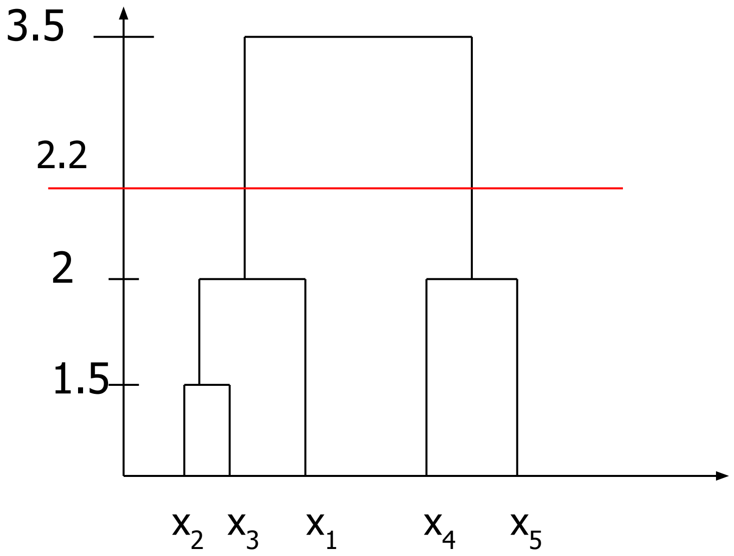
A Dendrogram Algorithm for Numerical variables

$D(x_1,x_2)$=2 D(x1,x3)=2.5 D(x1,x4)=5.39 D(x1,x5)=5

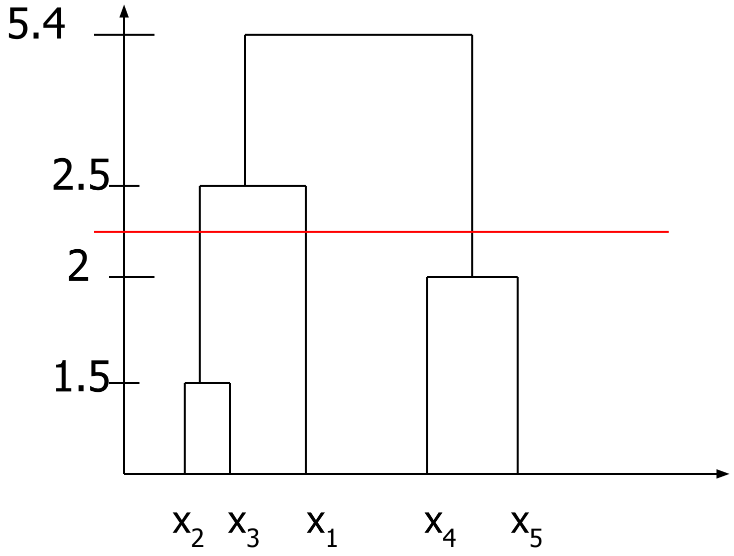
D(x2,x3)=1.5 D(x2,x4)=5 D(x2,x5)=5.29

D(x3,x4)=3.5 D(x3,x5)=4.03

D(x4,x5)=2



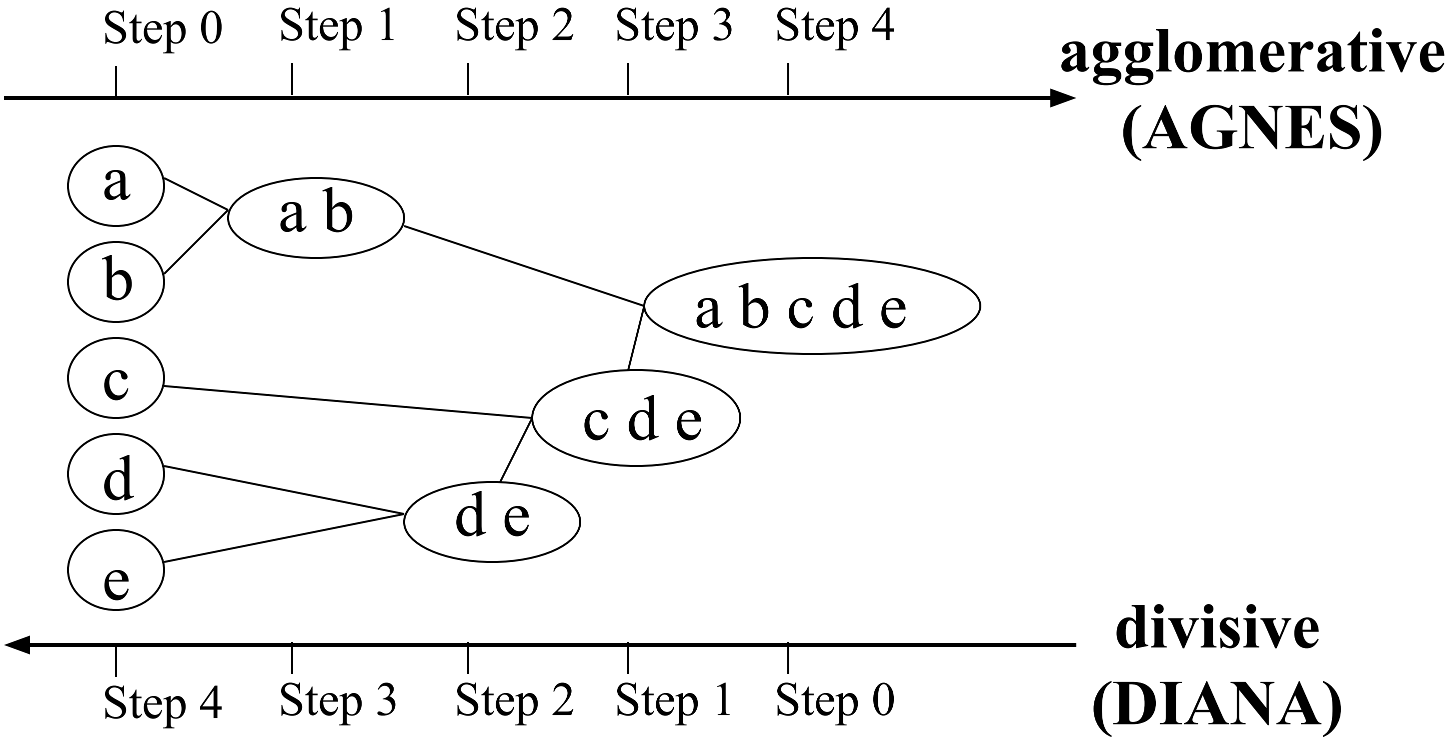
Dendrogram by single-link method



Dendrogram by complete-link method

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



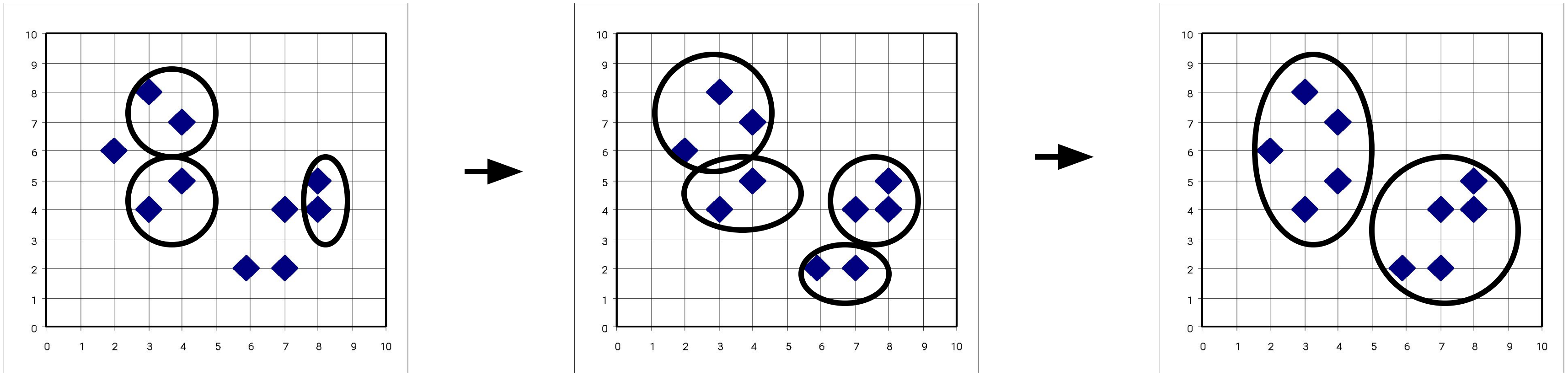
Agglomerative Hierarchical Clustering

1. Initially, each object forms its own cluster
2. Compute all pairwise distances between the initial clusters (objects)
3. Merge the closest pair (A, B) in the set of the current clusters into a new cluster C = A ∪ B
4. Remove A and B from the set of current clusters; insert C into the set of current clusters
5. If the set of current clusters contains only C (i.e., if C represents all objects from the database): STOP
6. Else: determine the distance between the new cluster C and all other clusters in the set of current clusters; go to step 3.

➢ Requires a distance function for clusters (sets of objects)

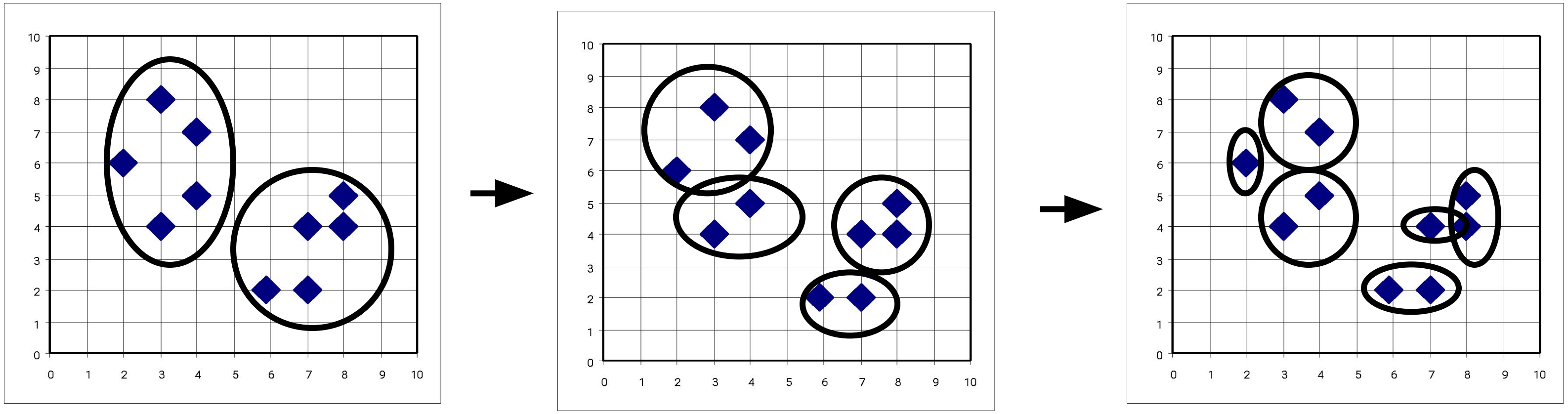
AGNES (Agglomerative Nesting)

- 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



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:* handles only numeric data, and sensitive to the order of the data record.

Clustering Feature Vector

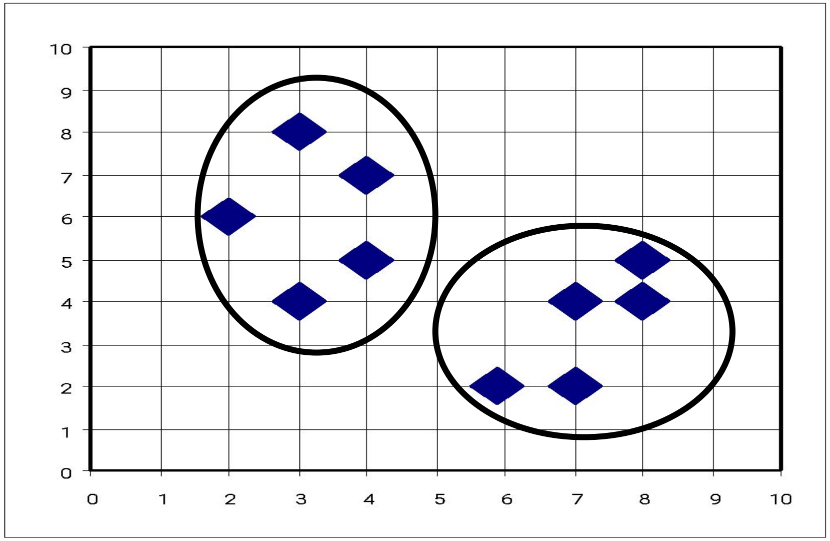
Clustering Feature: $CF = (N, \overrightarrow{LS}, SS)$

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)

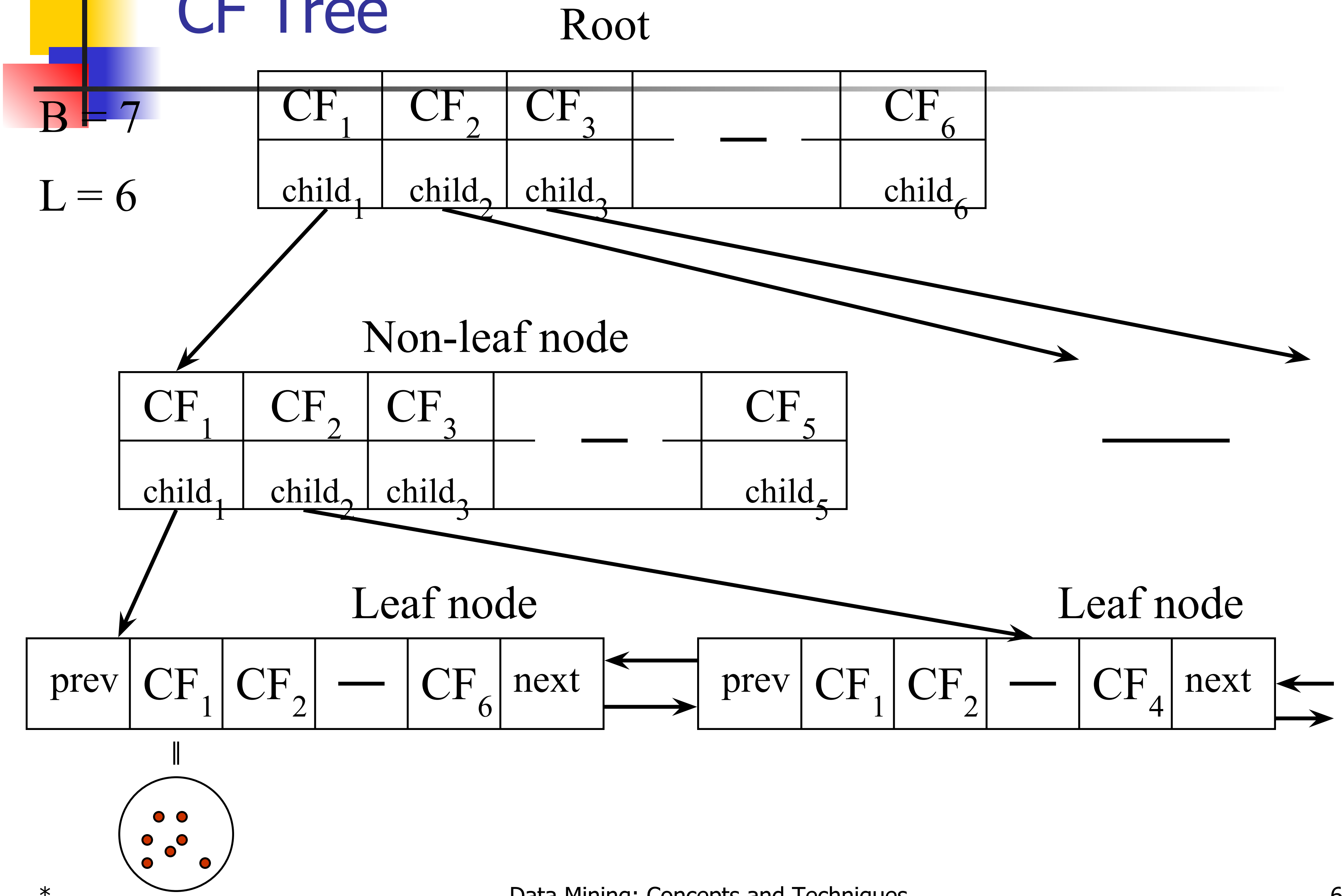
CF-Tree in BIRCH

- 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 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: specify the maximum number of children.
  - threshold: max diameter of sub-clusters stored at the leaf nodes

CF Tree

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
$child_1$	$child_2$	$child_3$		$child_5$

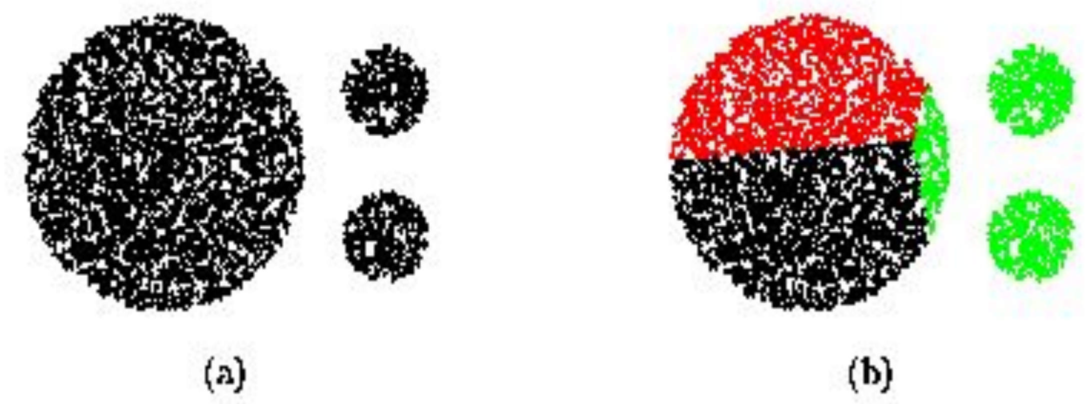
Leaf node

prev	CF_1	CF_2	—	CF_6	next

Leaf node

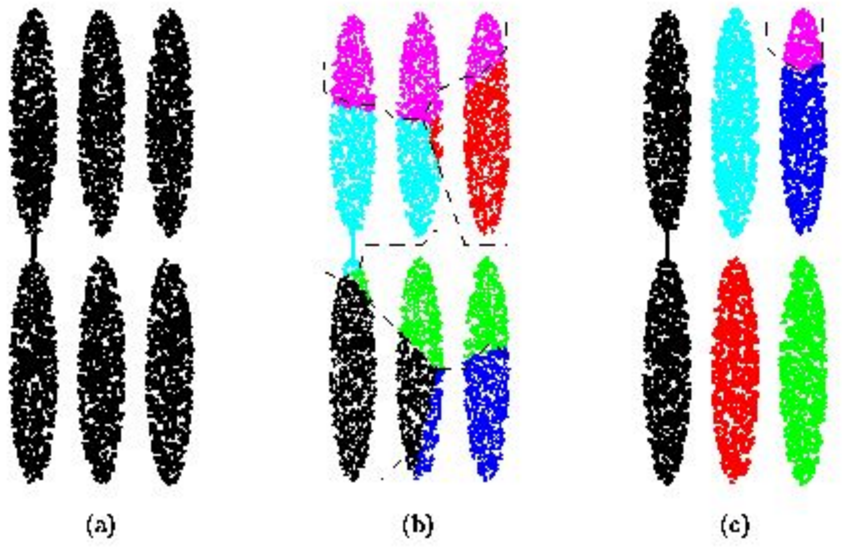
prev	CF_1	CF_2	—	CF_4	next

CURE (Clustering Using REpresentatives)



- 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



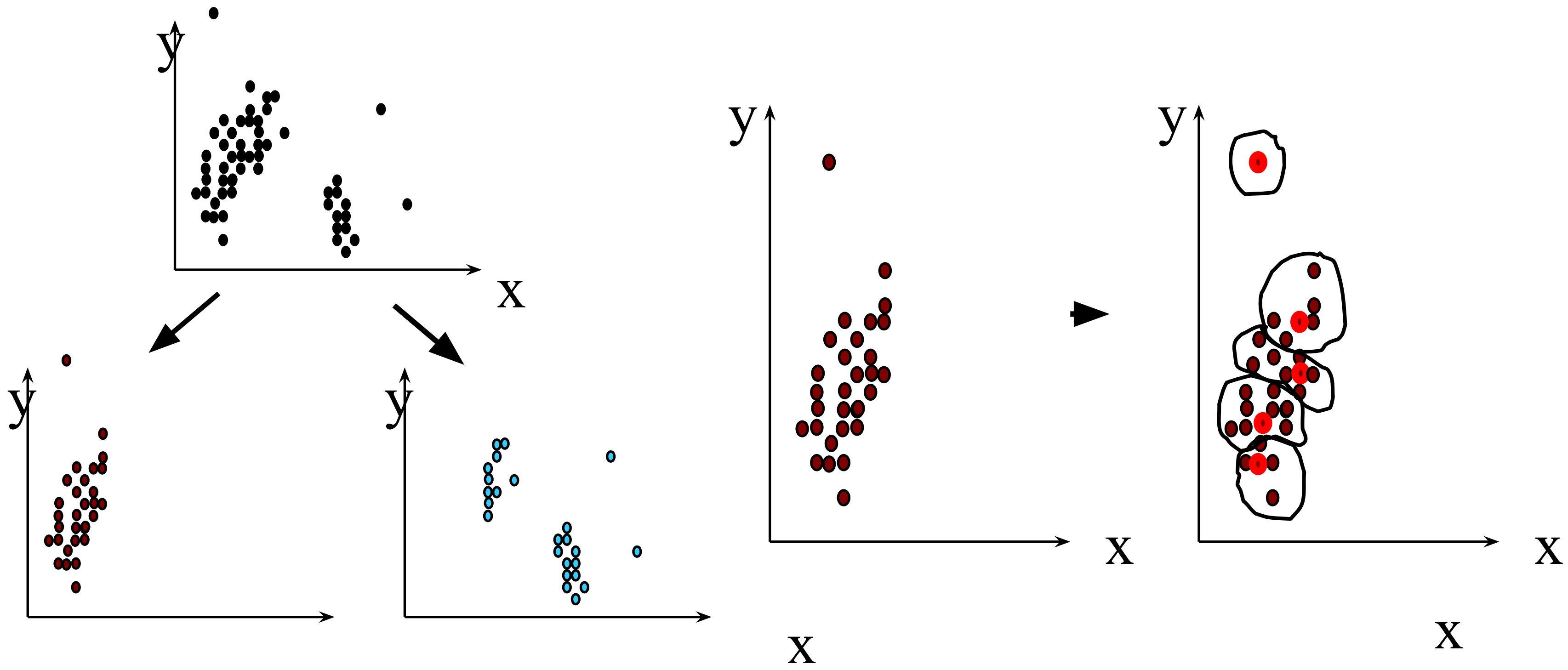
- Drawbacks of square-error based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex shaped, similar size and density, and if *k* can be reasonably estimated

Cure: The Algorithm

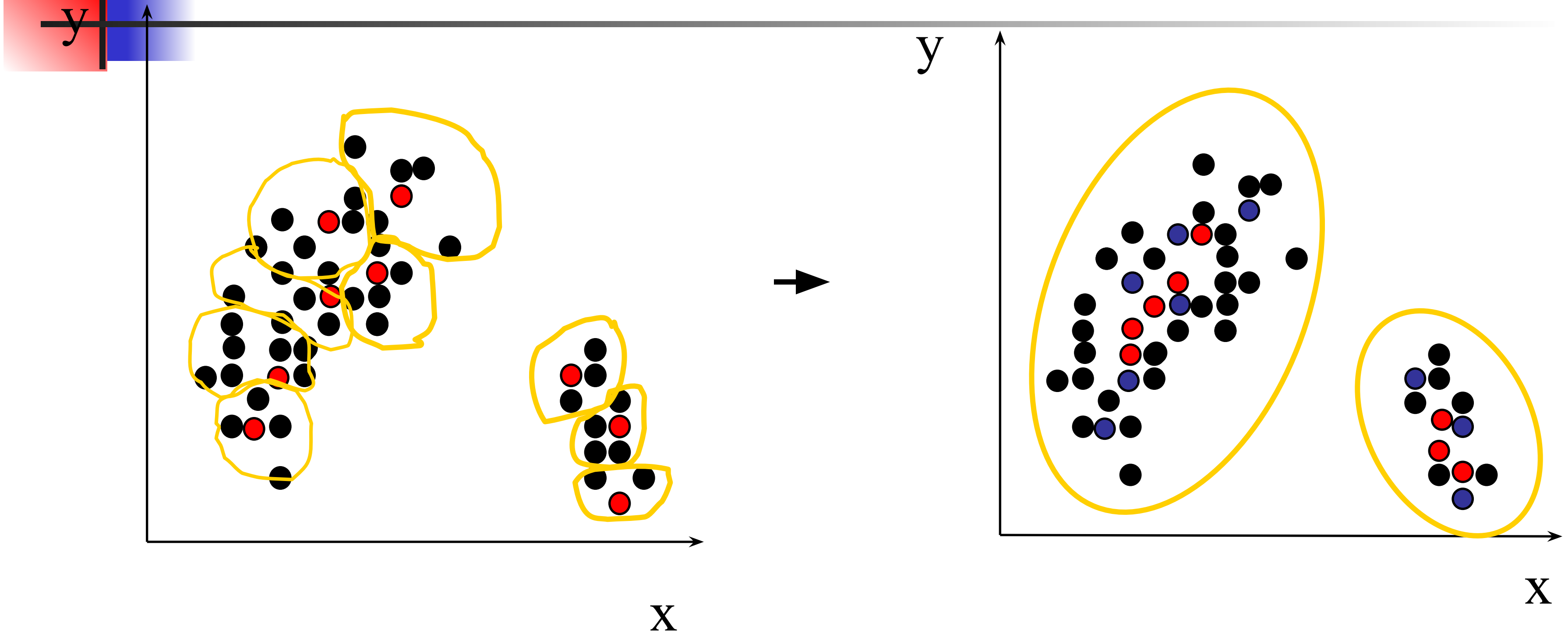
- Draw random sample *s*.
- Partition sample to *p* partitions with size *s/p*
- Partially cluster partitions into *s/pq* clusters
- Eliminate outliers
  - By random sampling
  - If a cluster grows too slow, eliminate it.
- Cluster partial clusters.

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 α.
- Multiple representatives capture the shape of the cluster

Clustering Categorical Data: ROCK

- ROCK: Robust Clustering using linKs,
  by S. Guha, R. Rastogi, K. Shim (ICDE'99).
  - Use links to measure similarity/proximity
  - Not distance based
  - Computational complexity: $O(n^2 + nm_m m_a + n^2 \log n)$
- Basic ideas:
  - Similarity function and neighbors: $Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$

    Let $T_1 = \{1,2,3\}$, $T_2=\{3,4,5\}$

$$Sim(T1, T2) = \frac{|\{3\}|}{|\{1,2,3,4,5\}|} = \frac{1}{5} = 0.2$$

Rock: Algorithm

- Links: The number of common neighbors for the two points.

{1,2,3}, {1,2,4}, {1,2,5}, {1,3,4}, {1,3,5}
{1,4,5}, {2,3,4}, {2,3,5}, {2,4,5}, {3,4,5}

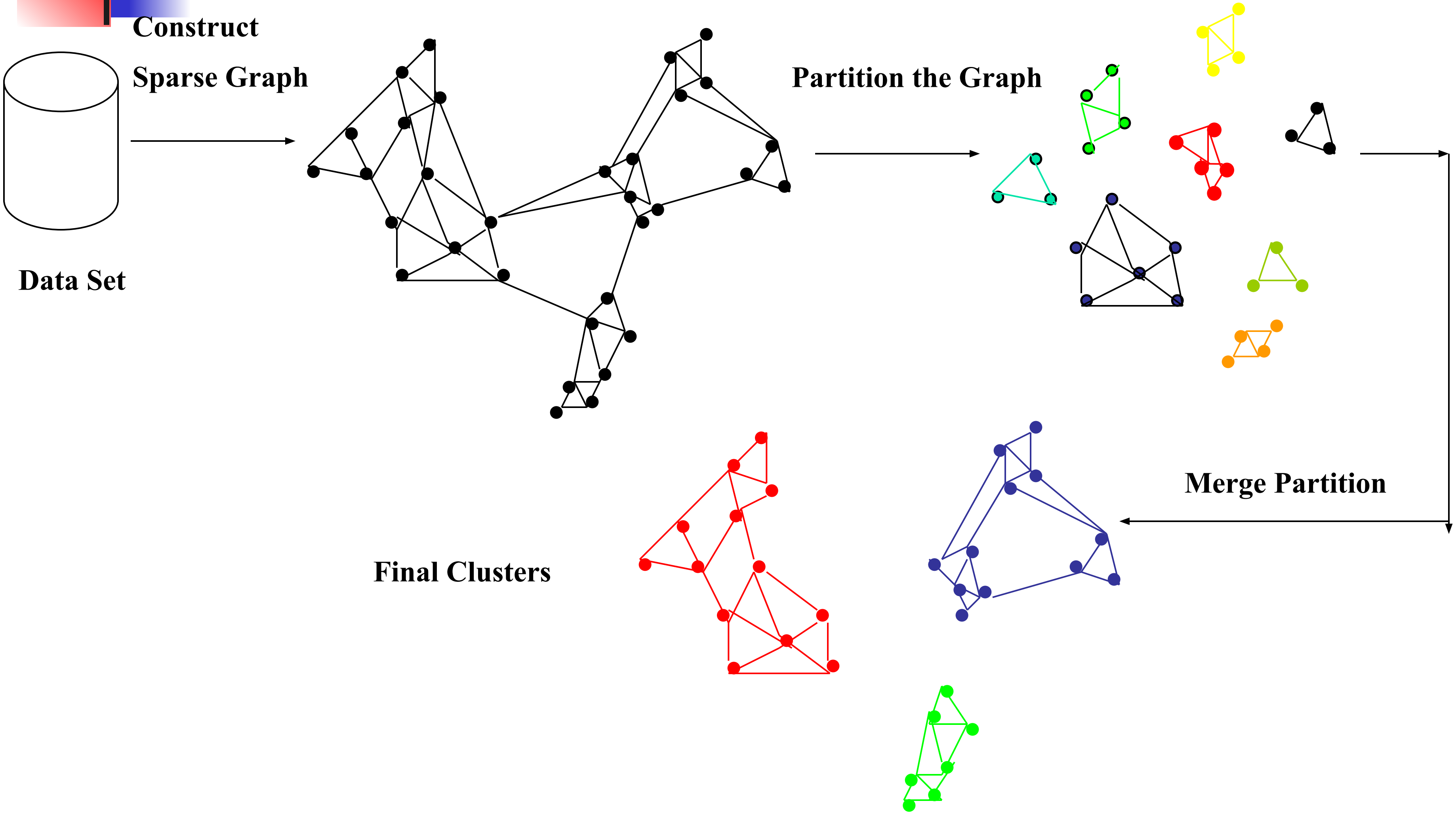
$$\{1,2,3\} \xleftrightarrow{3} \{1,2,4\}$$

- Algorithm
  - Draw random sample
  - Cluster with links

CHAMELEON (Hierarchical clustering using dynamic modeling)

- 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 closeness of items within the clusters
  - **Cure** ignores information about **interconnectivity** of the objects, **Rock** ignores information about the **closeness** of two 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



Construct

Sparse Graph

Data Set

Partition the Graph

Merge Partition

Final Clusters

Chapter 8. Cluster Analysis

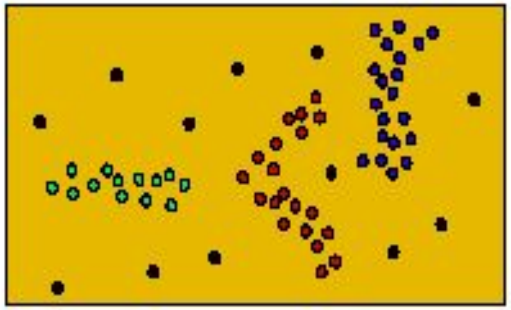
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Density-Based Clustering Methods

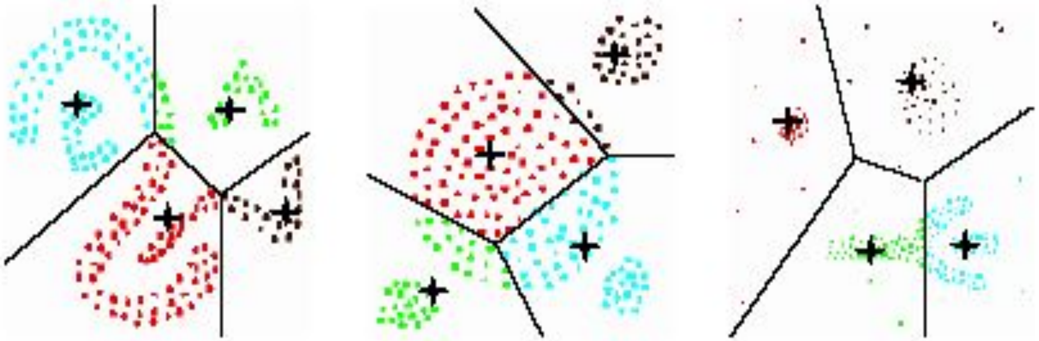
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Density-Based Clustering

- *Basic Idea:*
  - Clusters are dense regions in the data space, separated by regions of lower object density



- Why Density-Based Clustering?

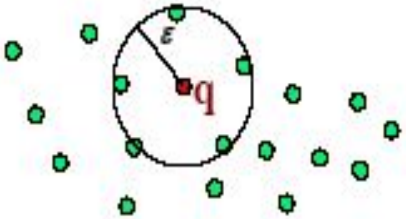


Results of a *k*-medoid algorithm for *k*=4

Different density-based approaches exist (see Textbook & Papers)
Here we discuss the ideas underlying the DBSCAN algorithm

Density Based Clustering: Basic Concept

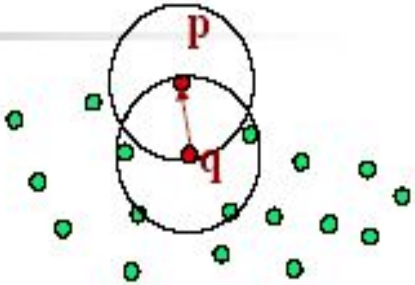
- Intuition for the formalization of the basic idea
  - For any point in a cluster, the local point density around that point has to exceed some threshold
  - The set of points from one cluster is spatially connected
- Local point density at a point *p* defined by two parameters
  - ε – radius for the neighborhood of point p:
    $N_{\varepsilon}(p) := \{q \text{ in data set } D \mid dist(p, q) \le \varepsilon\}$
  - ***MinPts*** – minimum number of points in the given neighbourhood *N*(*p*)
- *q* is called a ***core object*** (or core point) w.r.t. ε, *MinPts* if $| N_{\varepsilon}(q) | \ge MinPts$



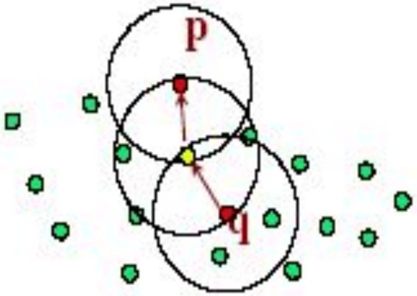
MinPts = 5 → **q** is a core object

Density Based Clustering: Basic Definitions

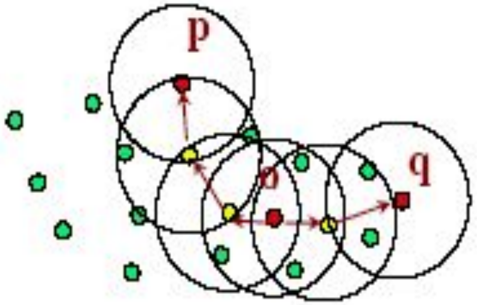
- *p* ***directly density-reachable*** from *q* w.r.t. *ε*, *MinPts* if
  1) $p \in N_\varepsilon(q)$ and
  2) *q* is a core object w.r.t. *ε*, *MinPts*



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



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



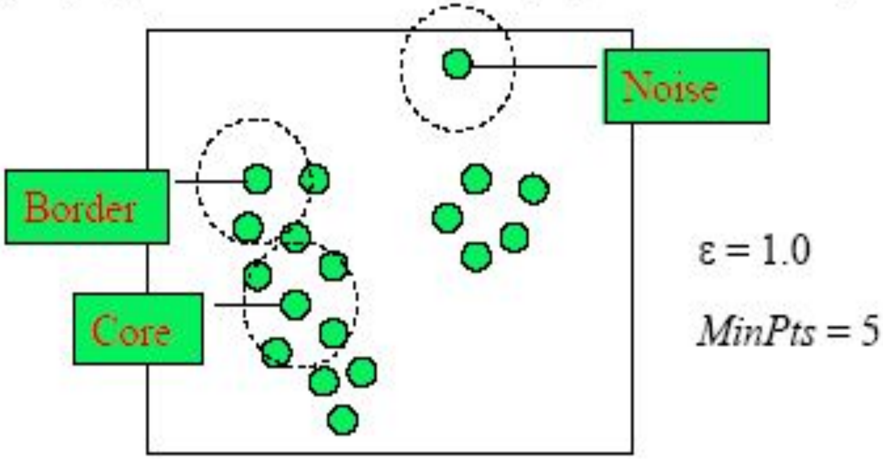
Density Based Clustering: Basic Definitions

- ***Density-Based Cluster***: non-empty subset *S* of database *D* satisfying:

  1) *Maximality*: if *p* is in *S* and *q* is density-reachable from *p* then *q* is in *S*

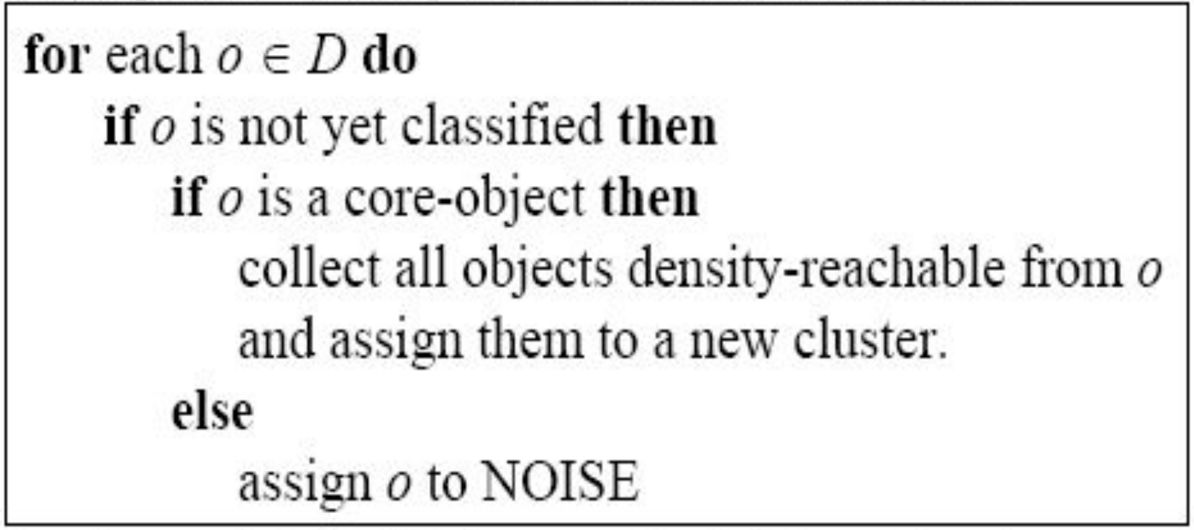
  2) *Connectivity*: each object in *S* is density-connected to all other objects

- ***Density-Based Clustering*** of a database *D* : $\{S_1, \ldots, S_n; N\}$ where
  - $S_1, \ldots, S_n$: all density-based clusters in the database *D*
  - $N = D \setminus \{S_1, \ldots, S_n\}$ is called the ***noise*** (objects not in any cluster)



Density Based Clustering: DBSCAN Algorithm

- Basic Theorem:
  - Each object in a density-based cluster C is density-reachable from any of its core-objects
  - Nothing else is density-reachable from core objects.

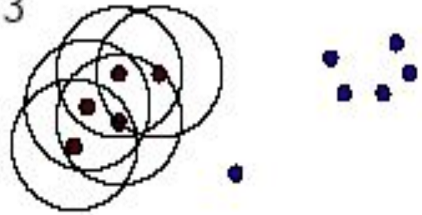


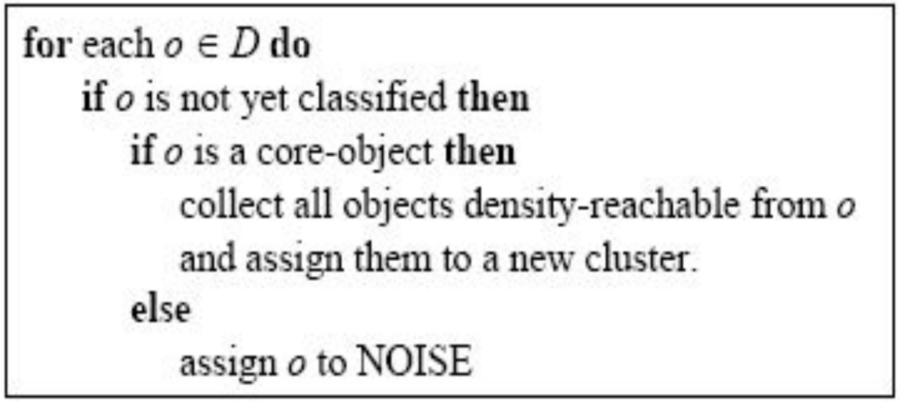
```
for each $o \in D$ do
    if $o$ is not yet classified then
        if $o$ is a core-object then
            collect all objects density-reachable from $o$
            and assign them to a new cluster.
        else
            assign $o$ to NOISE
```

  - density-reachable objects are collected by performing successive ε-neighborhood queries.

DBSCAN Algorithm: Example

- Parameter
  - $\varepsilon = 2.0$
  - $MinPts = 3$

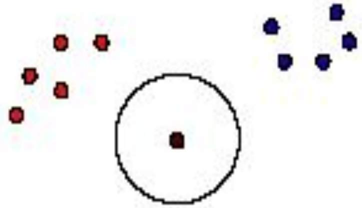




```
for each $o \in D$ do
    if $o$ is not yet classified then
        if $o$ is a core-object then
            collect all objects density-reachable from $o$
            and assign them to a new cluster.
        else
            assign $o$ to NOISE
```

DBSCAN Algorithm: Example

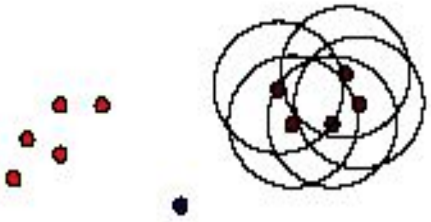
- Parameter
  - ε = 2.0
  - *MinPts* = 3

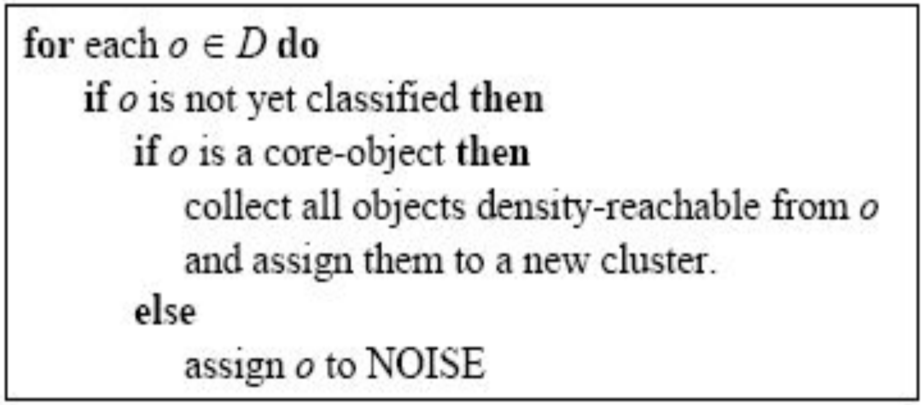


```
for each $o \in D$ do
    if $o$ is not yet classified then
        if $o$ is a core-object then
            collect all objects density-reachable from $o$
            and assign them to a new cluster.
        else
            assign $o$ to NOISE
```

DBSCAN Algorithm: Example

- Parameter
  - $\varepsilon = 2.0$
  - *MinPts* = 3

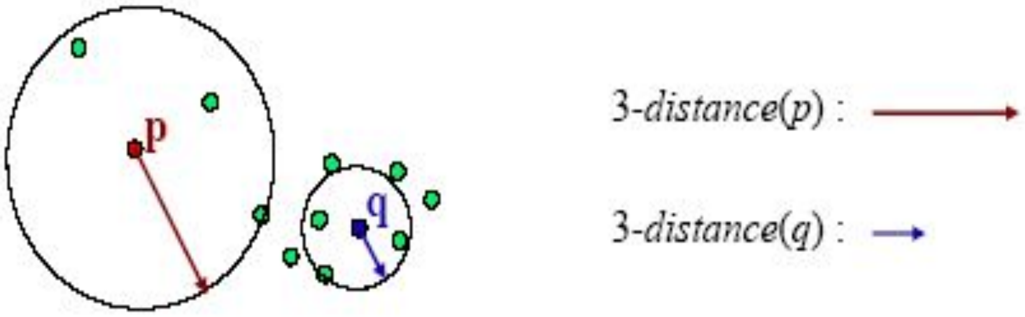




```
for each $o \in D$ do
    if $o$ is not yet classified then
        if $o$ is a core-object then
            collect all objects density-reachable from $o$
            and assign them to a new cluster.
        else
            assign $o$ to NOISE
```

Determining the Parameters ε and *MinPts*

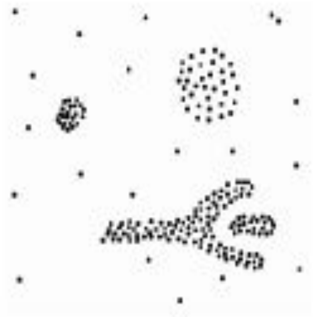
- Cluster: Point density higher than specified by ε and *MinPts*
- Idea: use the point density of the least dense cluster in the data set as parameters – but how to determine this?
- Heuristic: look at the distances to the *k*-nearest neighbors

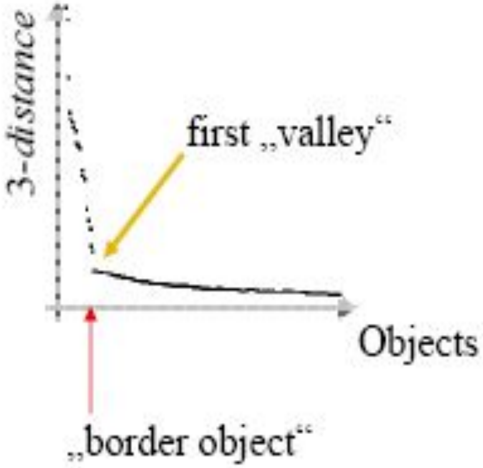


- Function *k-distance*(*p*): distance from *p* to the its *k*-nearest neighbor
- *k-distance plot*: *k*-distances of all objects, sorted in decreasing order

Determining the Parameters ε and *MinPts*

- Example *k*-distance plot





- Heuristic method:
  - Fix a value for *MinPts* (default: 2 × *d* −1)
  - User selects “border object” *o* from the *MinPts-distance* plot; ε is set to *MinPts-distance*(o)

Gradient: The steepness of a slope

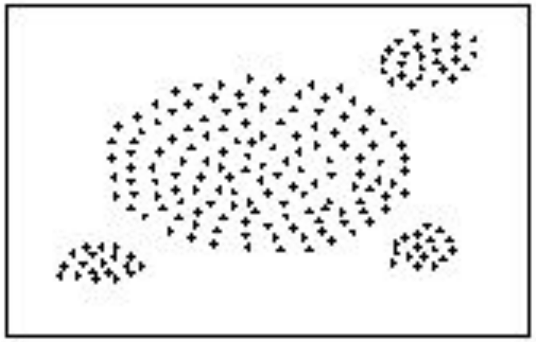
- Example

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

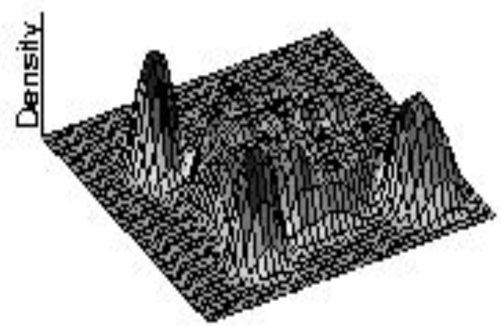
$$f_{Gaussian}^{D}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

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

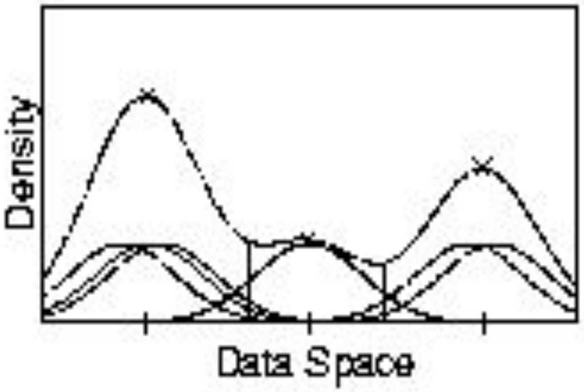
Density Attractor



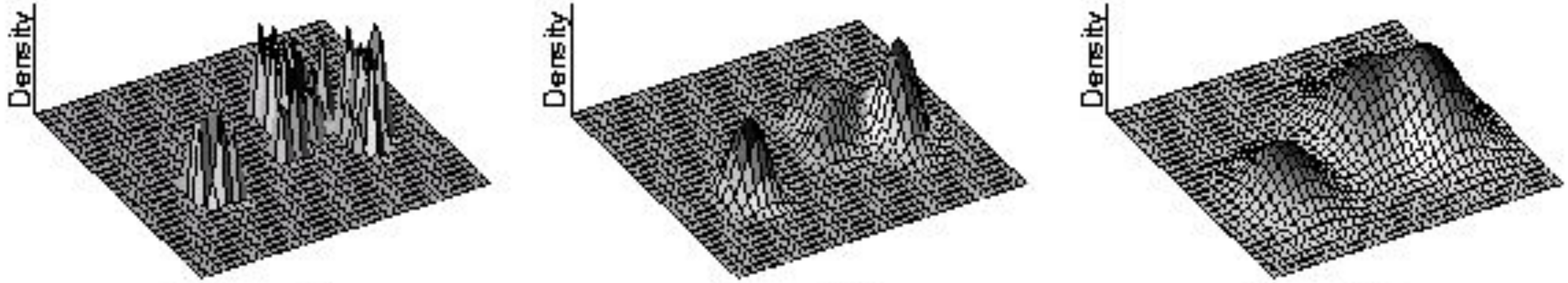
(a) Data Set



(c) Gaussian

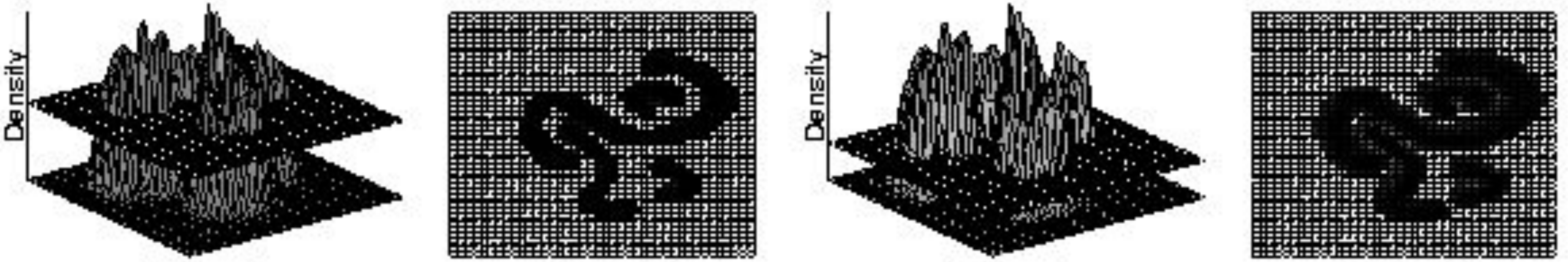


Center-Defined and Arbitrary



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

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

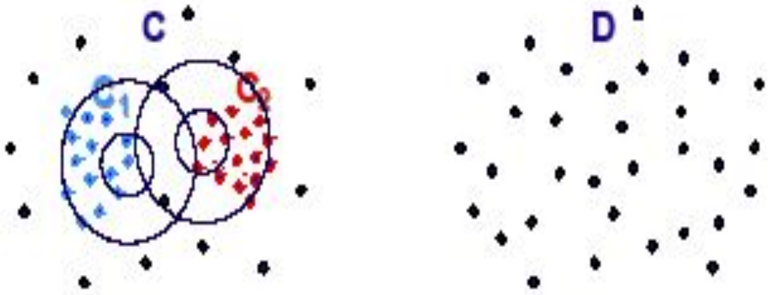


(a) $\xi = 2$ (b) $\xi = 2$ (c) $\xi = 1$ (d) $\xi = 1$

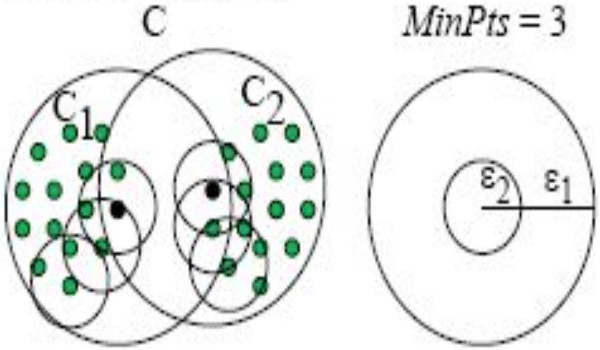
Figure 4: Example of Arbitray-Shape Clusters for different ξ

Density-Based Hierarchical Clustering

- *Observation*: Dense clusters are completely contained by less dense clusters



- *Idea*: Process objects in the "right" order and keep track of point density in their neighborhood



Core- and Reachability Distance

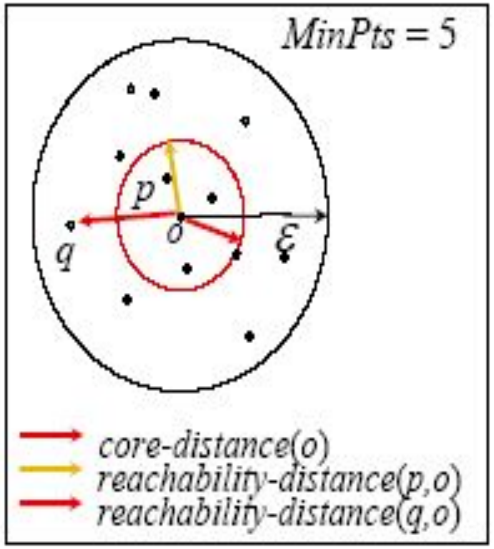
- Parameters: "generating" distance ε, fixed value *MinPts*
- ***core-distance***$_{\varepsilon, MinPts}$**(*o*)**

  "smallest distance such that *o* is a core object"

  (if that distance is $\leq \varepsilon$; "?" otherwise)
- ***reachability-distance***$_{\varepsilon, MinPts}$**(*p, o*)**

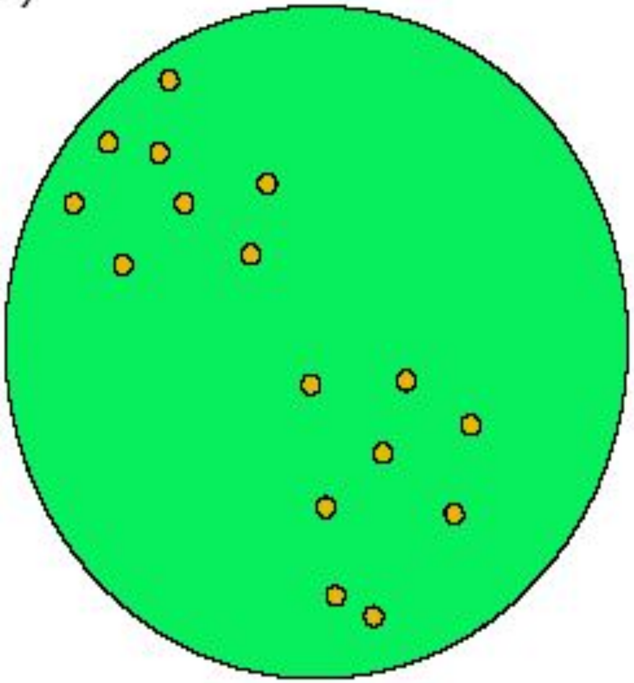
  "smallest distance such that *p* is *directly* density-reachable from *o*"

  (if that distance is $\leq \varepsilon$; "?" otherwise)



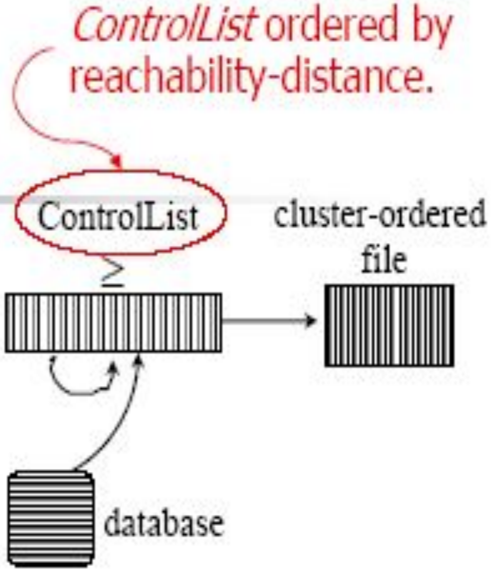
The Algorithm OPTICS

- Basic data structure: controlList
  - Memorize shortest reachability distances seen so far ("distance of a jump to that point")
- Visit each point
  - Make always a shortest jump
- Output:
  - order of points
  - core-distance of points
  - reachability-distance of points



The Algorithm OPTICS

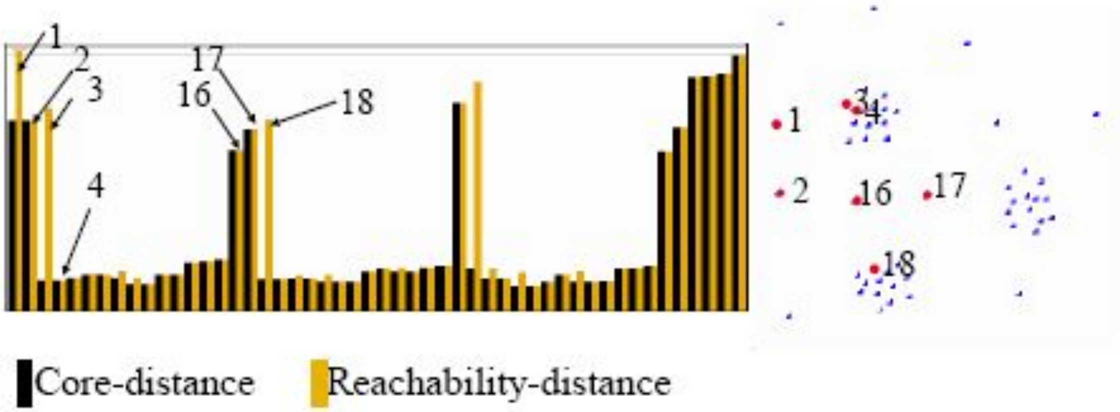
```
foreach $o \in$ Database
  // initially, $o$.processed = false for all objects $o$
  if $o$.processed = false;
    insert ($o$, "?") into ControlList;
  while ControlList is not empty
      select first element ($o$, $r$-$dist$) from ControlList;
      retrieve $N_\varepsilon(o)$ and determine $c\_dist$= core-distance($o$);
      set $o$.processed = true;
      write ($o$, $r\_dist$, $c\_dist$) to file;
      if $o$ is a core object at any distance $\leq \varepsilon$
        foreach $p \in N_\varepsilon(o)$ not yet processed;
            determine $r\_dist_p$ = reachability-distance($p$, $o$);
            if ($p$, _) $\notin$ ControlList
              insert ($p$, $r\_dist_p$) in ControlList;
            else if ($p$, $old\_r\_dist$) $\in$ ControlList and $r\_dist_p < old\_r\_dist$
              update ($p$, $r\_dist_p$) in ControlList;
```



ControlList ordered by reachability-distance.

OPTICS: Properties

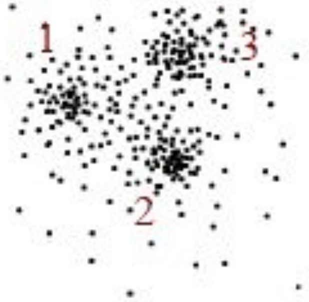
- “Flat” density-based clusters wrt. $\varepsilon^* \leq \varepsilon$ and *MinPts* afterwards:
  - Starts with an object *o* where *c-dist*(*o*) $\leq \varepsilon^*$ and *r-dist*(*o*) $> \varepsilon^*$
  - Continues while *r-dist* $\leq \varepsilon^*$

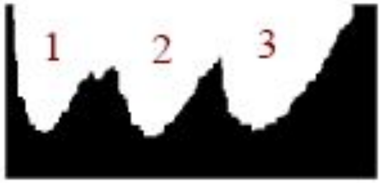


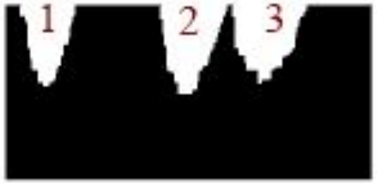
- Performance: approx. runtime(DBSCAN(ε, *MinPts*))
  - O(n * runtime(ε-neighborhood-query))
    - without spatial index support (worst case): O(n^2)
    - e.g. tree-based spatial index support: O($n * \log(n)$)

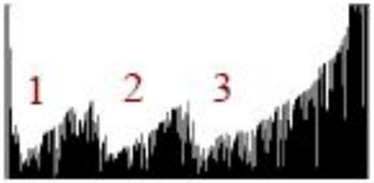
OPTICS: Parameter Sensitivity

- Relatively insensitive to parameter settings
- Good result if parameters are just "large enough"




MinPts = 10, ε = 10


MinPts = 10, **ε = 5**


***MinPts* = 2**, ε = 10

Chapter 8. Cluster Analysis

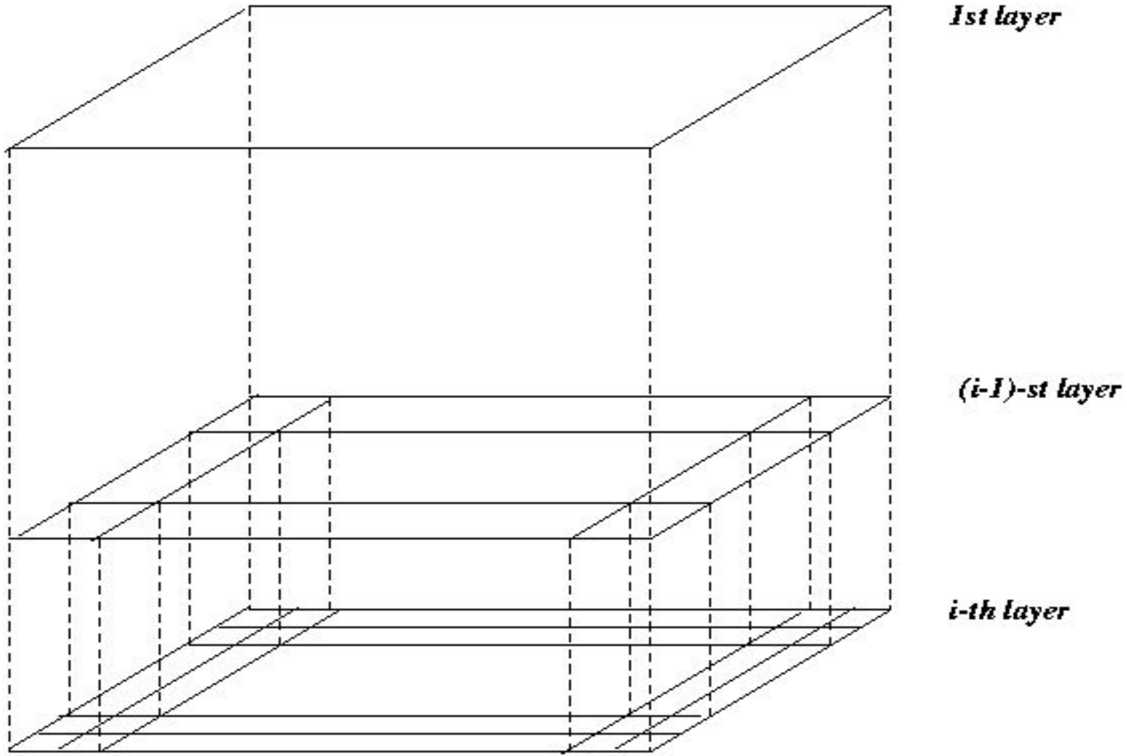
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

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

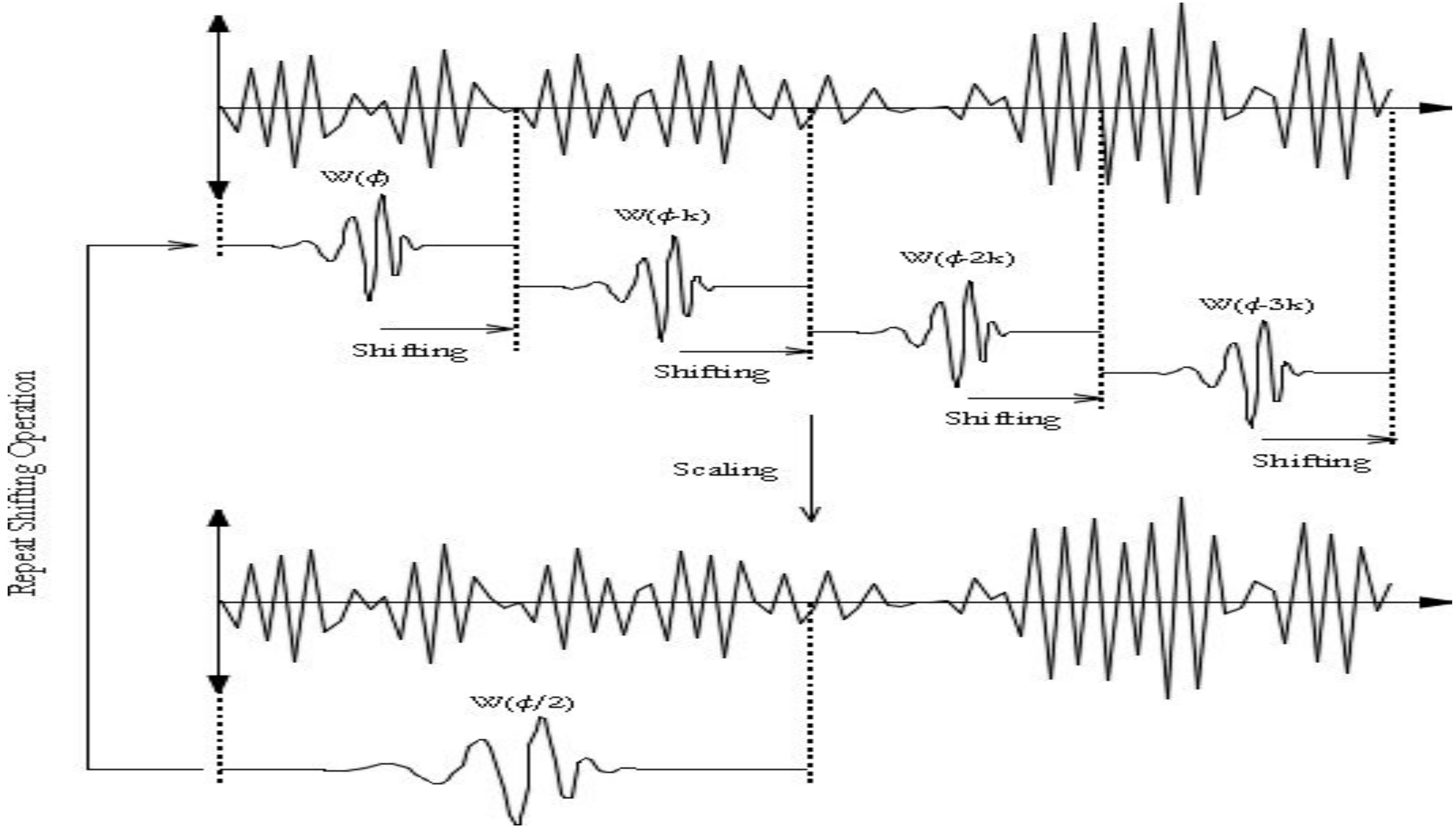
STING: A Statistical Information Grid Approach (3)

- 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
- 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

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.

What is Wavelet (1)?



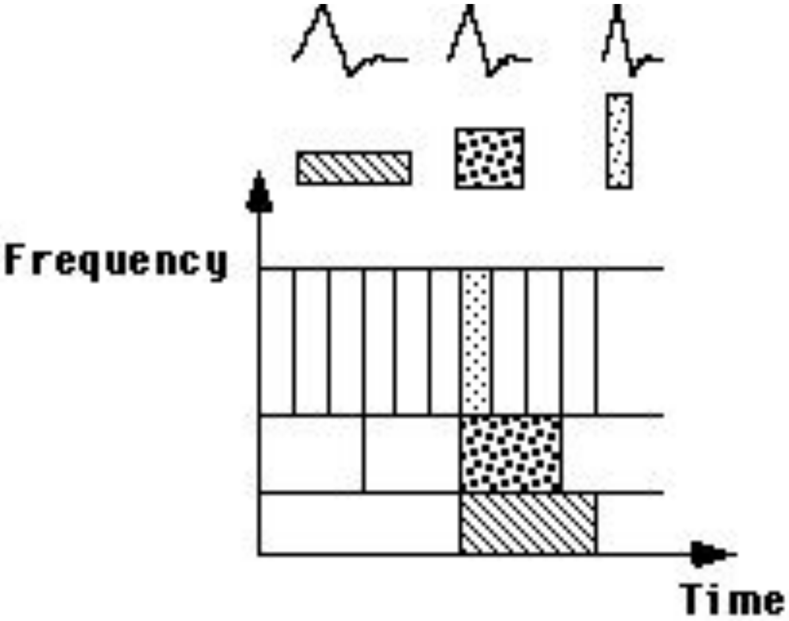
WaveCluster (1998)

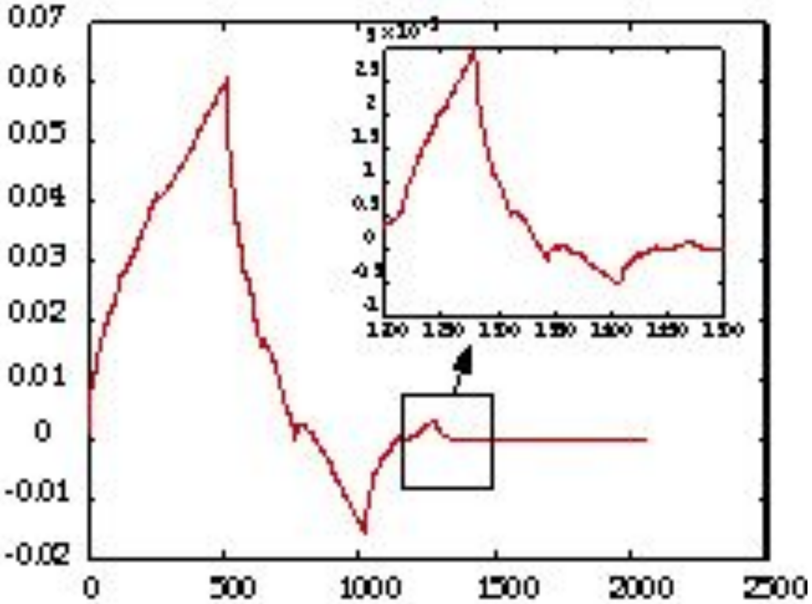
- How to apply wavelet transform to find clusters
  - Summaries the data by imposing a multidimensional grid structure onto data space
  - These multidimensional spatial data objects are represented in a n-dimensional feature space
  - Apply wavelet transform on feature space to find the dense regions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

Wavelet Transform

- Decomposes a signal into different frequency subbands. (can be applied to n-dimensional signals)
- Data are transformed to preserve relative distance between objects at different levels of resolution.
- Allows natural clusters to become more distinguishable

What Is Wavelet (2)?





Quantization

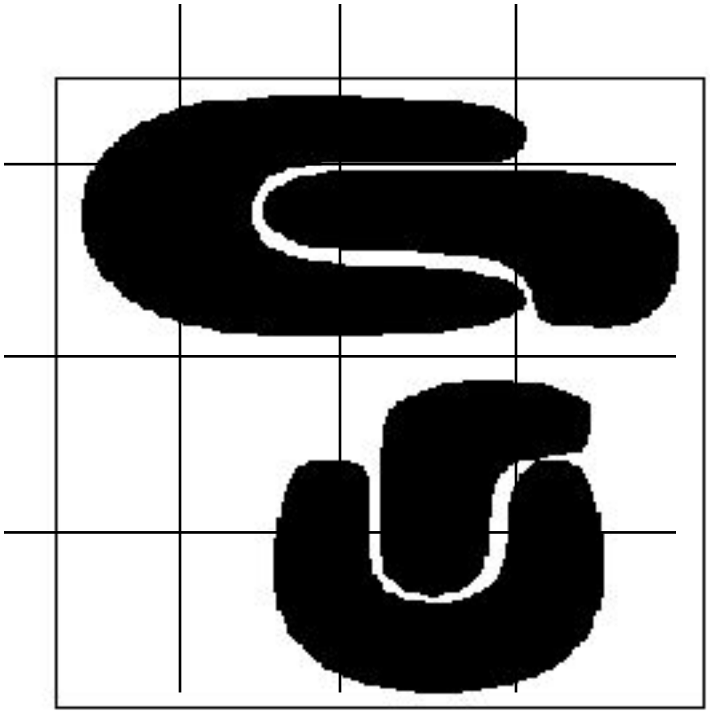
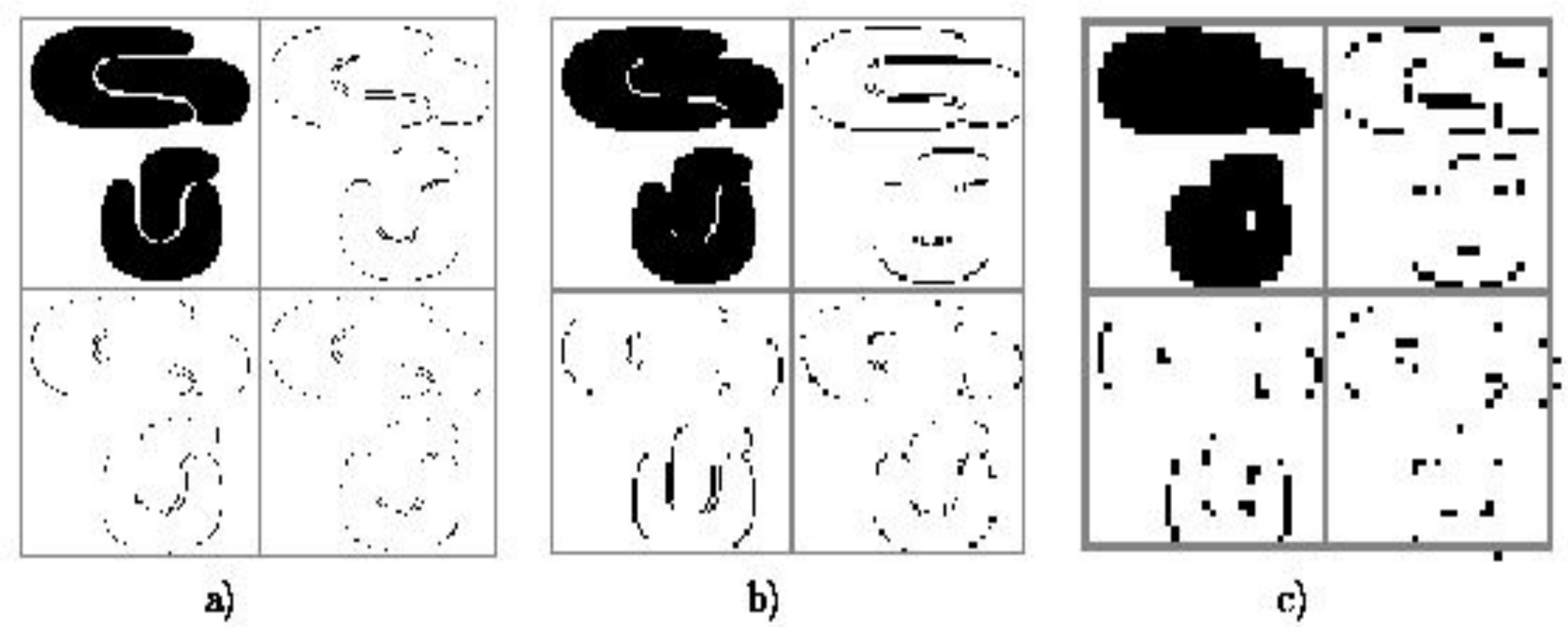


Figure 1: A sample 2-dimensional feature space.

Transformation



WaveCluster (1998)

- Why is wavelet transformation useful for clustering
  - Unsupervised clustering

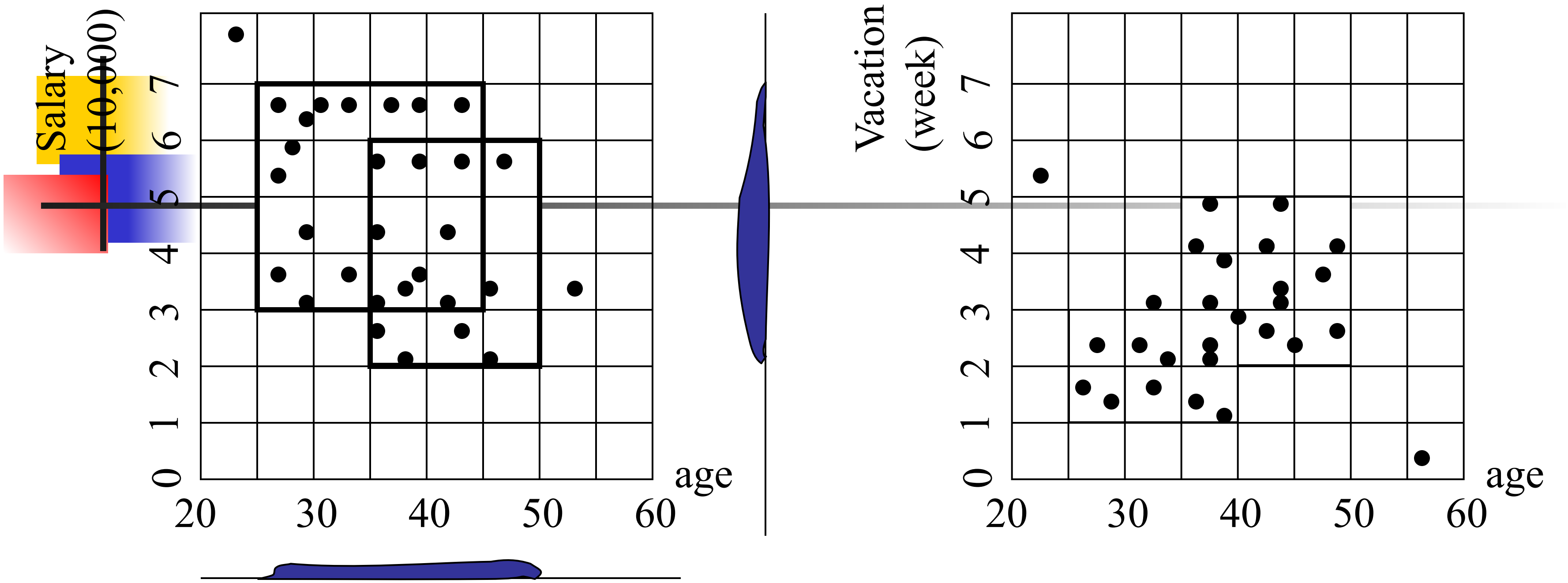
    It uses hat-shape filters to emphasize region where points cluster, but simultaneously to suppress weaker information in their boundary
  - Effective removal of outliers
  - Multi-resolution
  - Cost efficiency
- Major features:
  - Complexity O(N)
  - Detect arbitrary shaped clusters at different scales
  - Not sensitive to noise, not sensitive to input order
  - Only applicable to low dimensional data

CLIQUE (Clustering In QUEst)

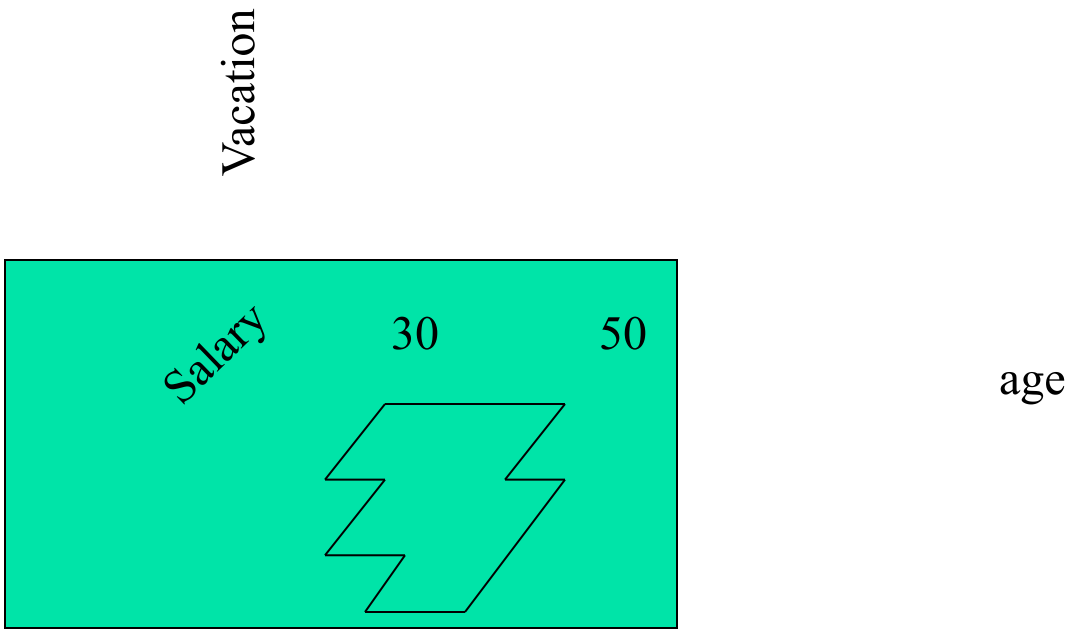
- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length 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

τ = 3


Vacation
Salary
30
50
age

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
  - It 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 8. Cluster Analysis

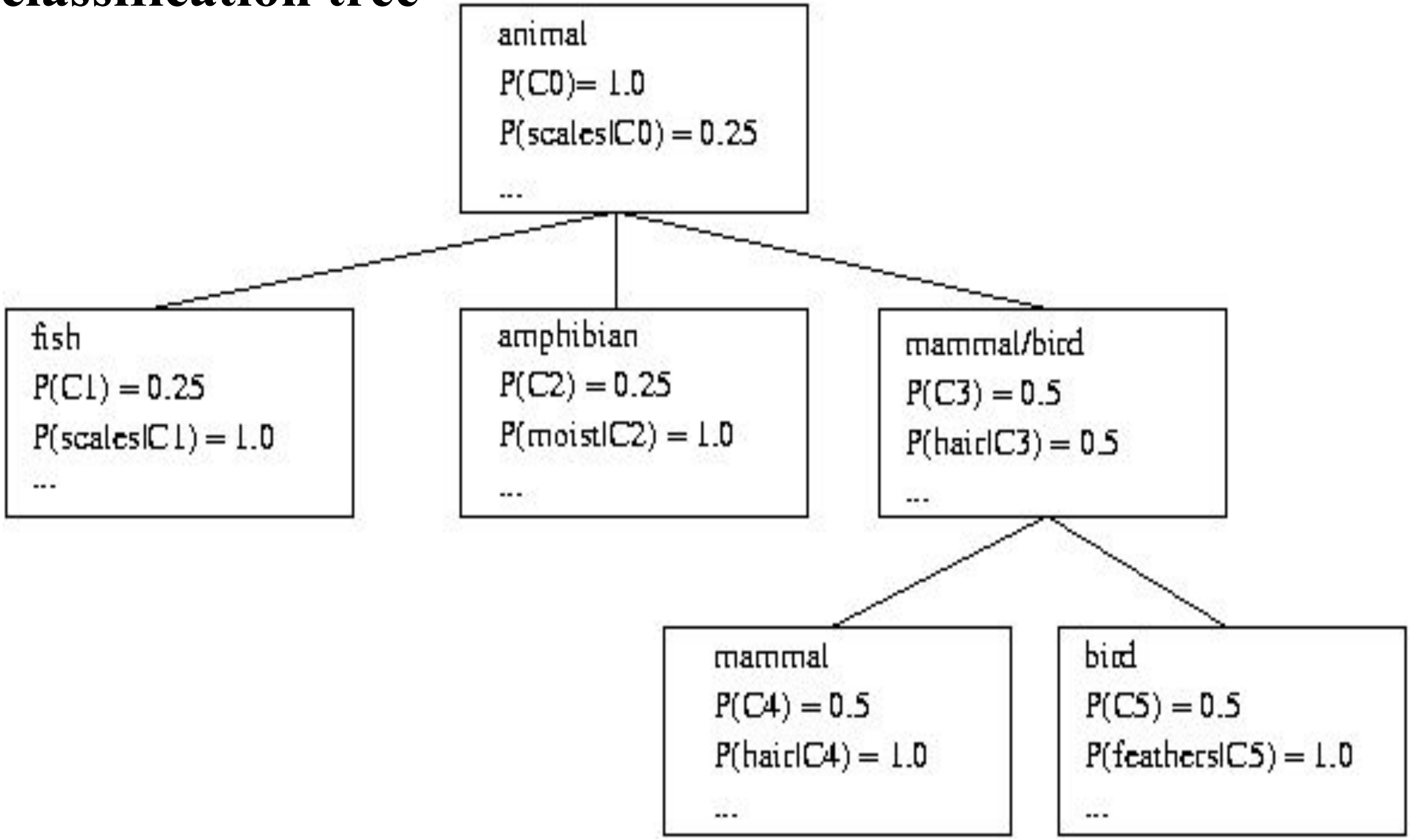
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Model-Based Clustering Methods

- Attempt to optimize the fit between the data and some mathematical model
- Statistical and AI approach
  - Conceptual clustering
    - A form of clustering in machine learning
    - Produces a classification scheme for a set of unlabeled objects
    - Finds characteristic description for each concept (class)
  - COBWEB (Fisher'87)
    - A popular a simple method of incremental conceptual learning
    - Creates a hierarchical clustering in the form of a classification tree
    - Each node refers to a concept and contains a probabilistic description of that concept

COBWEB Clustering Method

A classification tree



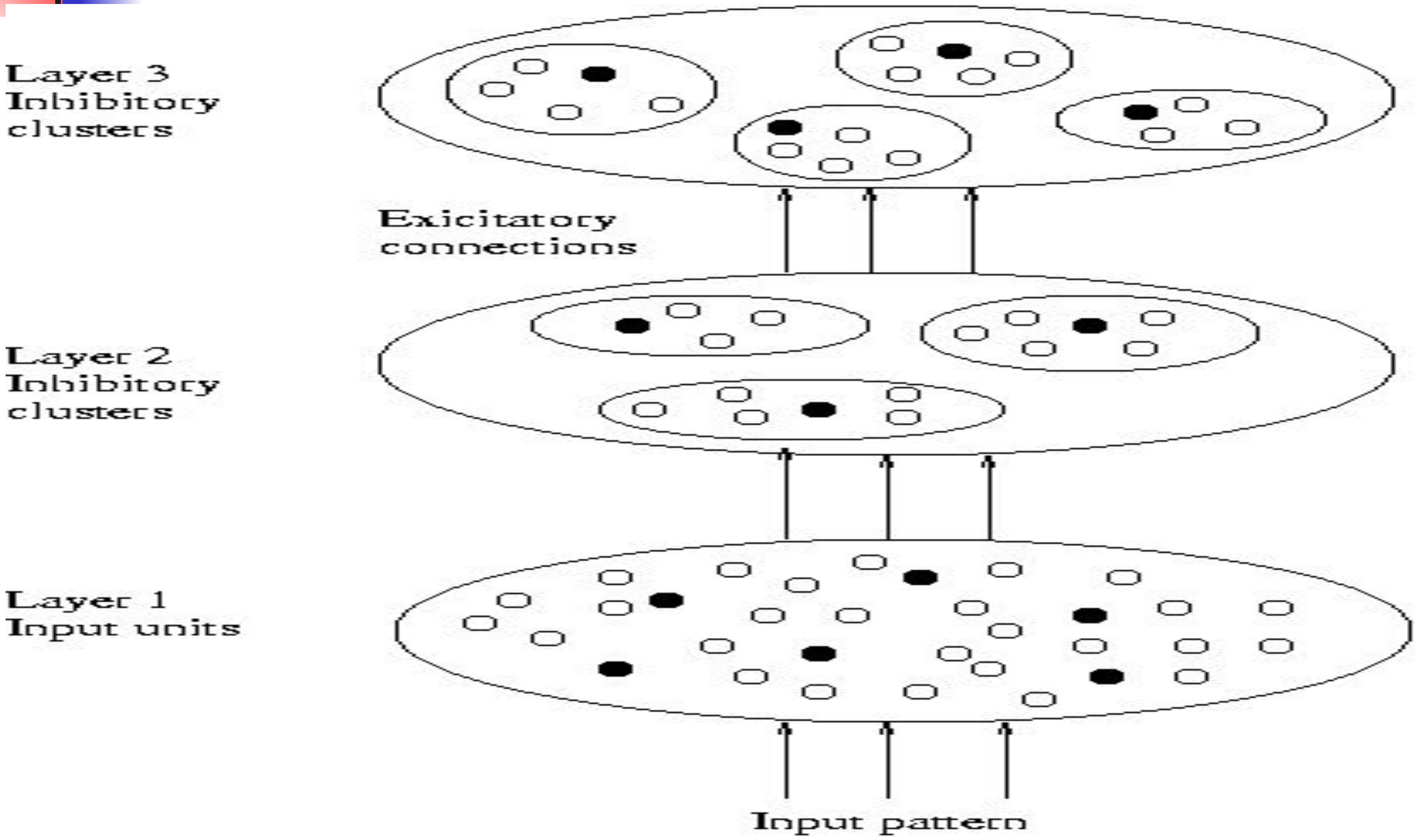
More on Statistical-Based Clustering

- Limitations of COBWEB
  - The assumption that the attributes are independent of each other is often too strong because correlation may exist
  - Not suitable for clustering large database data – skewed tree and expensive probability distributions
- CLASSIT
  - an extension of COBWEB for incremental clustering of continuous data
  - suffers similar problems as COBWEB
- AutoClass (Cheeseman and Stutz, 1996)
  - Uses Bayesian statistical analysis to estimate the number of clusters
  - Popular in industry

Other Model-Based Clustering Methods

- Neural network approaches
  - Represent each cluster as an exemplar, acting as a "prototype" of the cluster
  - New objects are distributed to the cluster whose exemplar is the most similar according to some dostance measure
- Competitive learning
  - Involves a hierarchical architecture of several units (neurons)
  - Neurons compete in a "winner-takes-all" fashion for the object currently being presented

Model-Based Clustering Methods



Self-organizing feature maps (SOMs)

- Clustering is also performed by having several units competing for the current object
- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

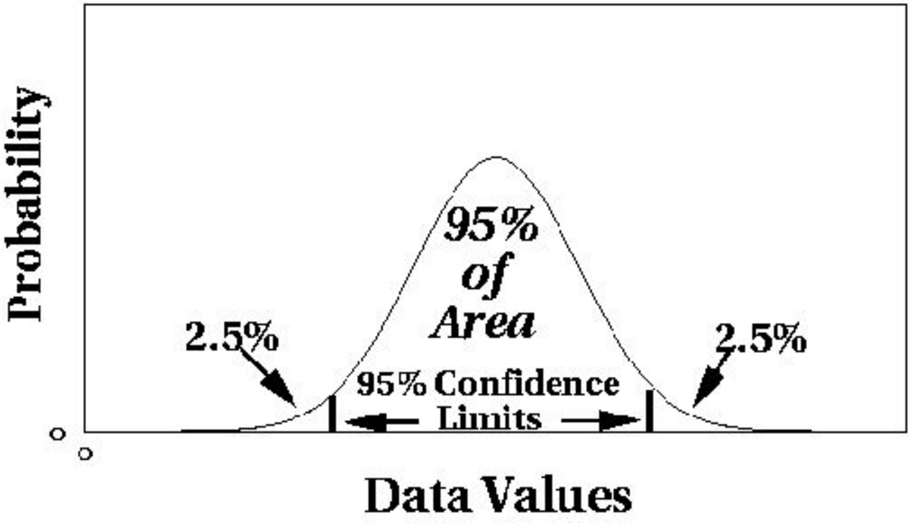
Chapter 8. Cluster Analysis

- 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
- Model-Based Clustering Methods
- 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 Approach



- 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

Chapter 8. Cluster Analysis

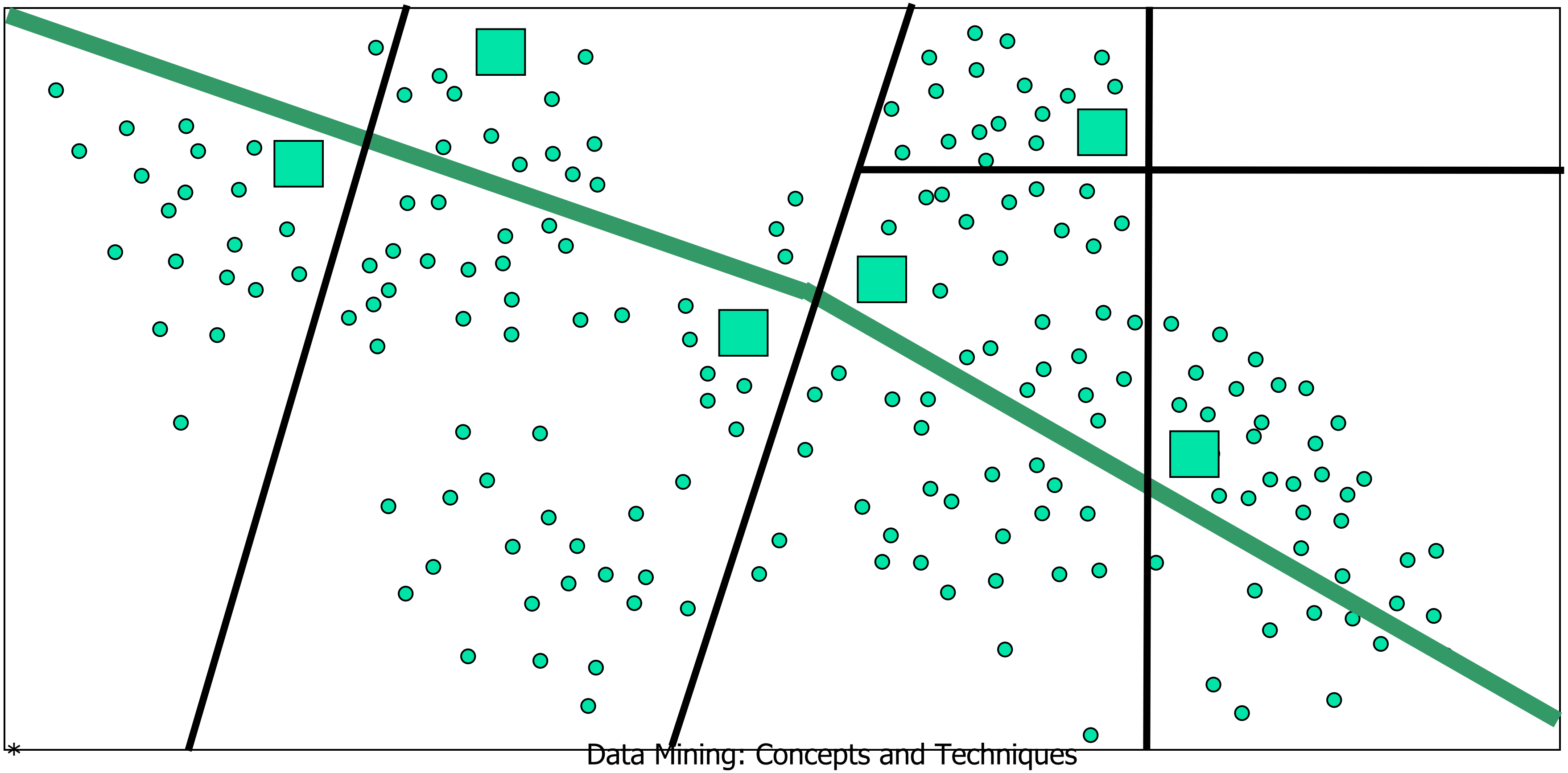
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Problems and Challenges

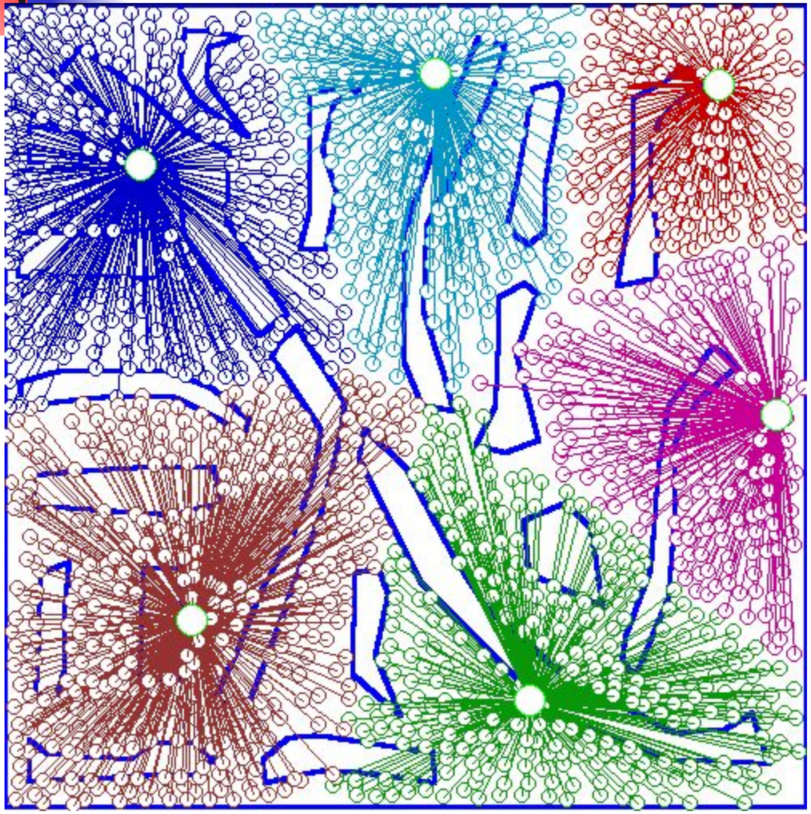
- Considerable progress has been made in scalable clustering methods
  - Partitioning: k-means, k-medoids, CLARANS
  - Hierarchical: BIRCH, CURE
  - Density-based: DBSCAN, CLIQUE, OPTICS
  - Grid-based: STING, WaveCluster
  - Model-based: Autoclass, Denclue, Cobweb
- Current clustering techniques do not address all the requirements adequately
- Constraint-based clustering analysis: Constraints exist in data space (bridges and highways) or in user queries

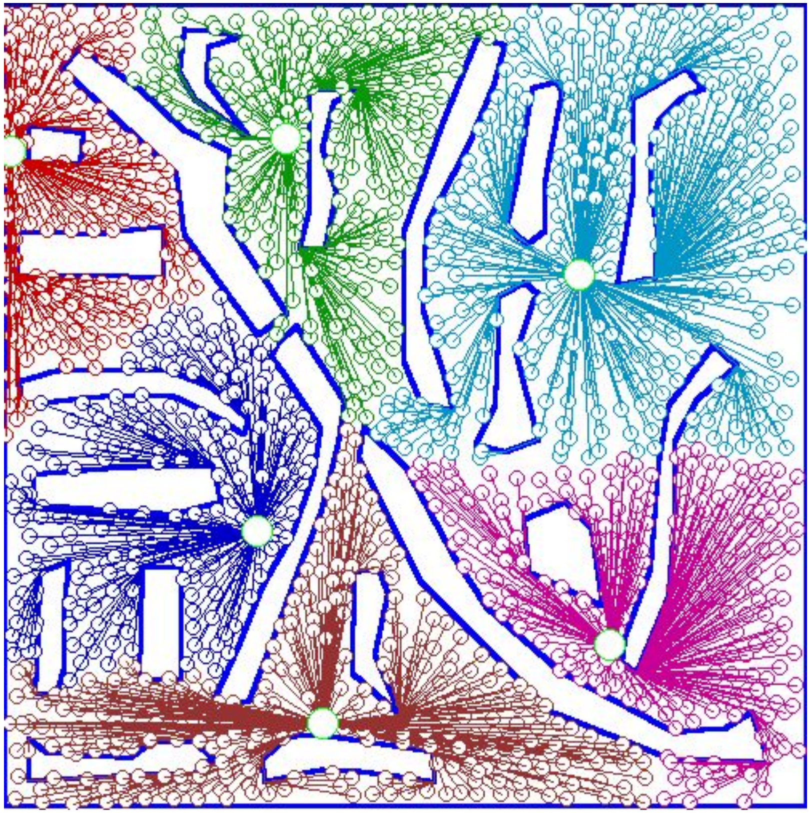
Constraint-Based Clustering Analysis

- Clustering analysis: less parameters but more user-desired constraints, e.g., an ATM allocation problem



Clustering With Obstacle Objects





Not Taking obstacles into account   Taking obstacles into account

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, such as constraint-based clustering

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD’99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scietific, 1996
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB’98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.

References (2)

- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB’98.
- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. Future Generation Computer systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition, 101-105.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB’98.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB’97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.