

# Cluster Analysis



# 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
- ❑ Clustering is used:
  - As a **stand-alone tool** to get insight into data distribution
    - ❑ Visualization of clusters may unveil important information
  - As a **preprocessing step** for other algorithms
    - ❑ Efficient indexing or compression often relies on clustering

# 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
  - cluster images based on their visual content
- ❑ Economic Science (especially market research)
- ❑ WWW and IR
  - document classification
  - cluster Weblog data to discover groups of similar access patterns

# 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.

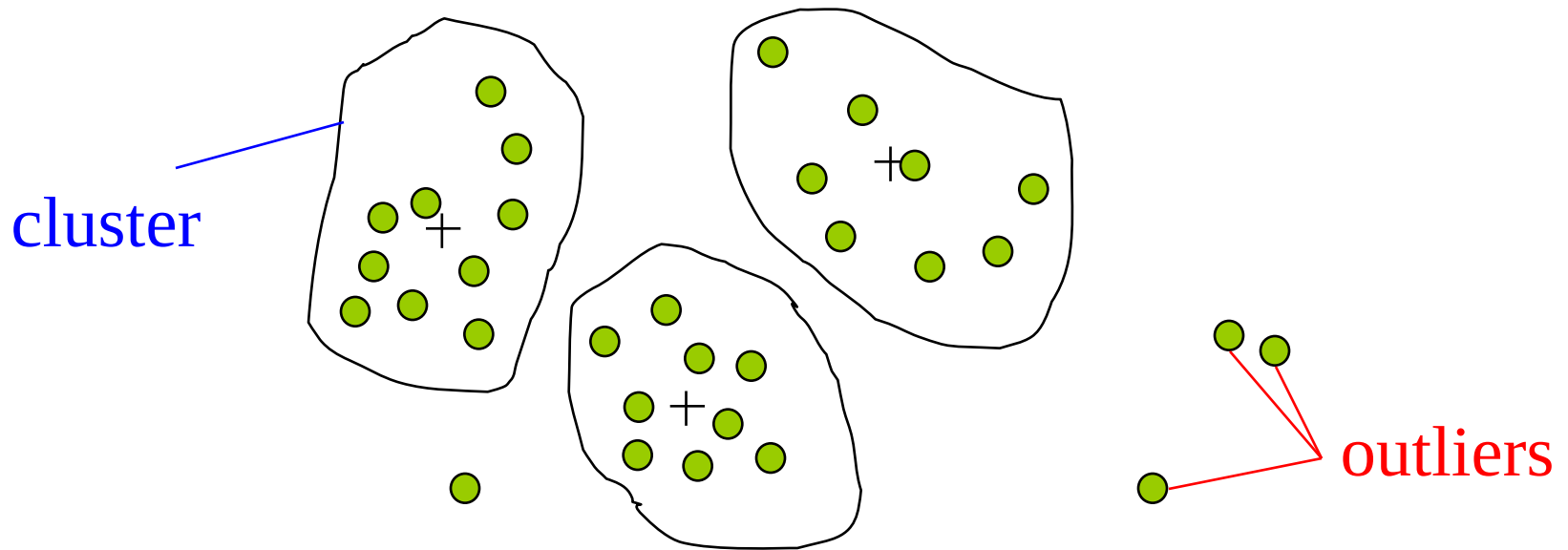
# Requirements of Clustering in Data Mining

---

- ❑ 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

# Outliers

- Outliers are objects that do not belong to any cluster or form clusters of very small cardinality



- In some applications we are interested in discovering outliers, not clusters (**outlier analysis**)

# 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)

the “classic” data input

|          |         |          |         |          |
|----------|---------|----------|---------|----------|
| $x_{11}$ | $\dots$ | $x_{1f}$ | $\dots$ | $x_{1p}$ |
| $\dots$  | $\dots$ | $\dots$  | $\dots$ | $\dots$  |
| $x_{i1}$ | $\dots$ | $x_{if}$ | $\dots$ | $x_{ip}$ |
| $\dots$  | $\dots$ | $\dots$  | $\dots$ | $\dots$  |
| $x_{n1}$ | $\dots$ | $x_{nf}$ | $\dots$ | $x_{np}$ |

- *dissimilarity* or *distance* matrix

- (one mode)

the desired data input to some clustering algorithms

|          |          |          |         |   |
|----------|----------|----------|---------|---|
| 0        |          |          |         |   |
| $d(2,1)$ | 0        |          |         |   |
| $d(3,1)$ | $d(3,2)$ | 0        |         |   |
| $\vdots$ | $\vdots$ | $\vdots$ |         |   |
| $d(n,1)$ | $d(n,2)$ | $\dots$  | $\dots$ | 0 |

# Measuring Similarity in Clustering

---

- Dissimilarity/Similarity metric:
  - The dissimilarity  $d(i, j)$  between two objects  $i$  and  $j$  is expressed in terms of a **distance function**, which is typically a **metric**:
    - $d(i, j) \geq 0$  (**non-negativity**)
    - $d(i, i) = 0$  (**isolation**)
    - $d(i, j) = d(j, i)$  (**symmetry**)
    - $d(i, j) \leq d(i, h) + d(h, j)$  (**triangular inequality**)
- The definitions of distance functions are usually different for **interval-scaled**, **boolean**, **categorical**, **ordinal** and **ratio-scaled** variables.
- Weights may be associated with different variables based on applications and data semantics.

# Type of data in cluster analysis

---

- Interval-scaled variables
  - e.g., salary, height
- Binary variables
  - e.g., gender (M/F), has\_cancer(T/F)
- Nominal (categorical) variables
  - e.g., religion (Christian, Muslim, Buddhist, Hindu, etc.)
- Ordinal variables
  - e.g., military rank (soldier, sergeant, lutenant, captain, etc.)
- Ratio-scaled variables
  - population growth (1,10,100,1000,...)
- Variables of mixed types
  - multiple attributes with various types

# Similarity and Dissimilarity Between Objects

---

- Distance metrics are normally used to measure the similarity or dissimilarity between two data objects
- The most popular conform to *Minkowski distance*:

$$L_p(i, j) = \left( |x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \dots + |x_{in} - x_{jn}|^p \right)^{1/p}$$

where  $i = (x_{i1}, x_{i2}, \dots, x_{in})$  and  $j = (x_{j1}, x_{j2}, \dots, x_{jn})$  are two  $n$ -dimensional data objects, and  $p$  is a positive integer

- If  $p = 1$ ,  $L_1$  is the *Manhattan (or city block)* distance:

$$L_1(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{in} - x_{jn}|$$

# Similarity and Dissimilarity Between Objects (Cont.)

---

□ If  $p = 2$ ,  $L_2$  is the Euclidean distance:

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_n} - x_{j_n}|^2)}$$

## ■ 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:

$$d(i, j) = \sqrt{(w_1 |x_{i_1} - x_{j_1}|^2 + w_2 |x_{i_2} - x_{j_2}|^2 + \dots + w_n |x_{i_n} - x_{j_n}|^2)}$$

# Binary Variables

- A binary variable has two states: 0 absent, 1 present

- A contingency table for binary data

|                 |            | object <i>j</i> |            |            |
|-----------------|------------|-----------------|------------|------------|
|                 |            | 1               | 0          | <i>sum</i> |
| object <i>i</i> | 1          | <i>a</i>        | <i>b</i>   | <i>a+b</i> |
|                 | 0          | <i>c</i>        | <i>d</i>   | <i>c+d</i> |
|                 | <i>sum</i> | <i>a+c</i>      | <i>b+d</i> | <i>p</i>   |

- Simple matching coefficient distance (invariant, if the binary variable is symmetric):

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

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

# Binary Variables

- Another approach is to define the **similarity** of two objects and not their **distance**.
- In that case we have the following:
  - Simple **matching coefficient** similarity:

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

- **Jaccard coefficient** similarity:

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

Note that:  $s(i, j) = 1 - d(i, j)$

# Dissimilarity between Binary Variables

---

## □ Example (Jaccard coefficient)

| Name | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|------|-------|-------|--------|--------|--------|--------|
| Jack | 1     | 0     | 1      | 0      | 0      | 0      |
| Mary | 1     | 0     | 1      | 0      | 1      | 0      |
| Jim  | 1     | 1     | 0      | 0      | 0      | 0      |

- all attributes are asymmetric binary
- 1 denotes presence or positive test
- 0 denotes absence or negative test

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

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

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

# A simpler definition

---

- Each variable is mapped to a bitmap (binary vector)

| Name | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|------|-------|-------|--------|--------|--------|--------|
| Jack | 1     | 0     | 1      | 0      | 0      | 0      |
| Mary | 1     | 0     | 1      | 0      | 1      | 0      |
| Jim  | 1     | 1     | 0      | 0      | 0      | 0      |

- Jack: 101000
- Mary: 101010
- Jim: 110000

- Simple match distance:

$$d(i, j) = \frac{\text{number of non - common bit positions}}{\text{total number of bits}}$$

- Jaccard coefficient:

$$d(i, j) = 1 - \frac{\text{number of 1's in } i \wedge j}{\text{number of 1's in } i \vee j}$$

# Variables of Mixed Types

---

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

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

# 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 random partitions and then iteratively refine them by some criterion
- ❑ Hierarchical 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

# 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 Concepts

---

- ❑ 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

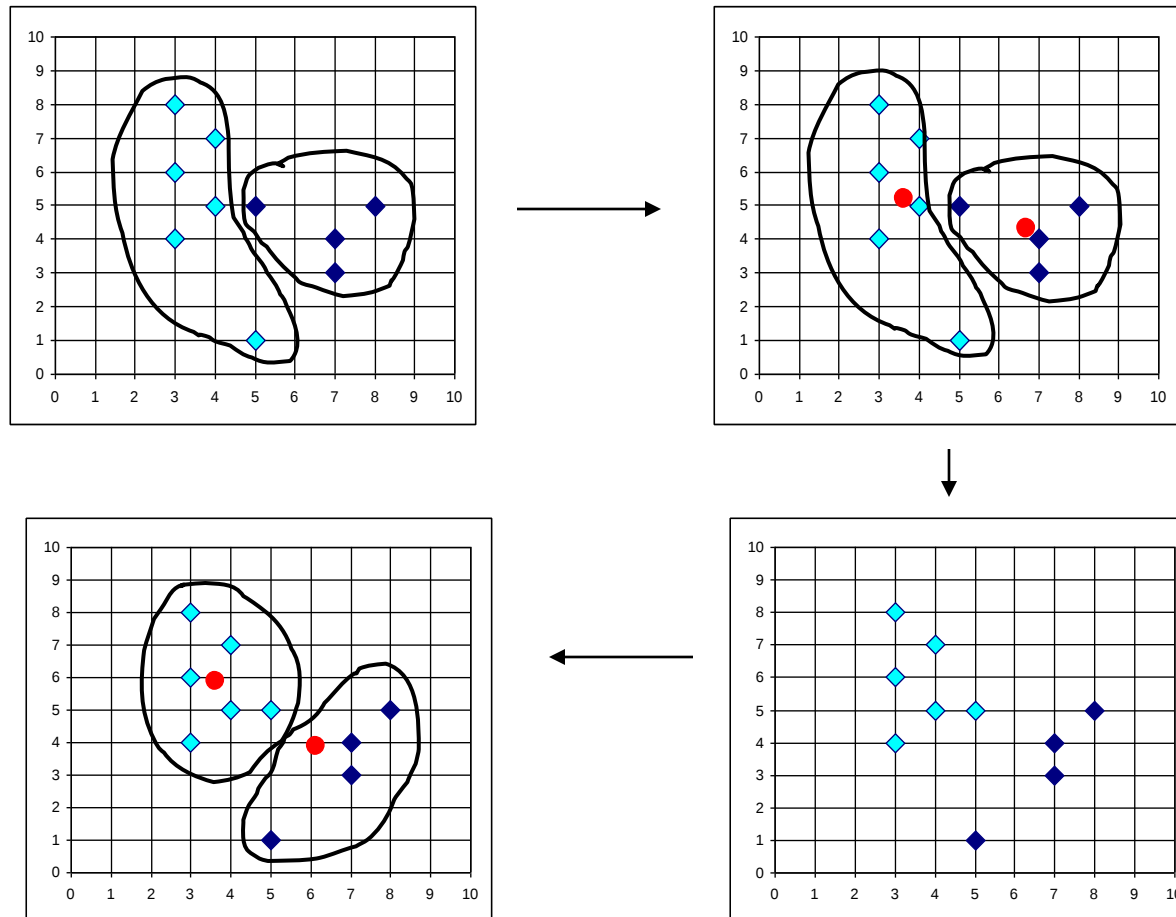
# The $k$ -means Clustering Method

---

- Given  $k$ , the  $k$ -means algorithm is implemented in 4 steps:
  1. Partition objects into  $k$  nonempty subsets
  2. Compute seed points as the **centroids** of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each object to the cluster with the nearest seed point.
  4. Go back to Step 2, stop when no more new assignment.

# The k-means Clustering Method

## □ Example



# Comments on the k-means Method

---

## □ Strength

- *Relatively efficient:  $O(tkn)$ , where  $n$  is # objects,  $k$  is # clusters, and  $t$  is # iterations. Normally,  $k, t \ll n$ .*
- Often terminates at a *local optimum*.

## □ Weaknesses

- 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*

# 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

# PAM (Partitioning Around Medoids) (1987)

---

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

# PAM Clustering: Total swapping cost

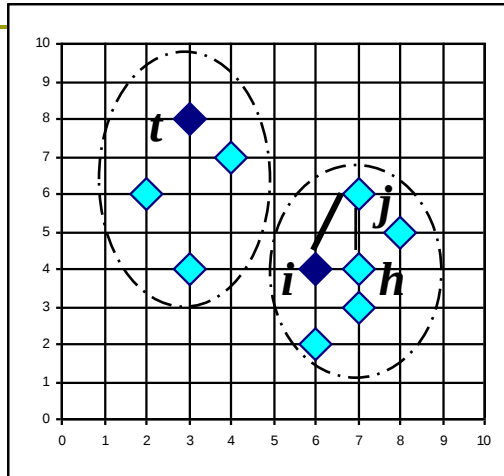
$$TC_{ih} = \sum_j C_{jih}$$

---

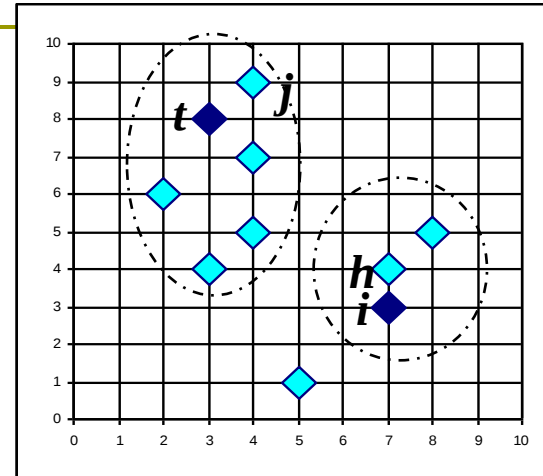
- $i$  is a current medoid,  $h$  is a non-selected object
- Assume that  $i$  is replaced by  $h$  in the set of medoids
- $TC_{ih} = 0$ ;
- For each non-selected object  $j \neq h$ :
  - $TC_{ih} += d(j, \text{new\_med}_j) - d(j, \text{prev\_med}_j)$ :
    - $\text{new\_med}_j$  = the closest medoid to  $j$  after  $i$  is replaced by  $h$
    - $\text{prev\_med}_j$  = the closest medoid to  $j$  before  $i$  is replaced by  $h$

# PAM Clustering: Total swapping cost

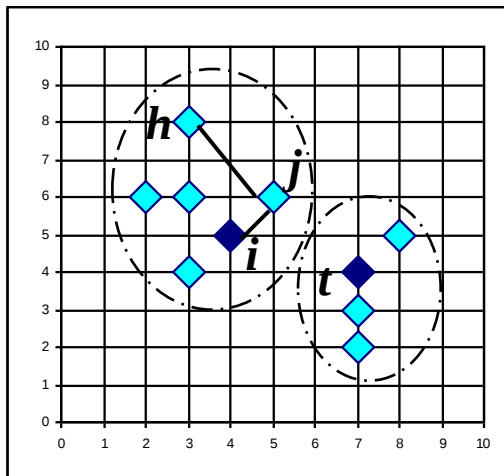
$$TC_{ib} = \sum_j C_{jih}$$



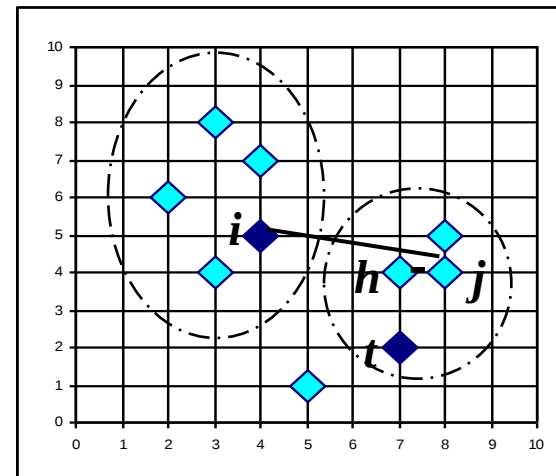
$$C_{jih} = d(j, h) - d(j, i)$$



$$C_{jih} = 0$$



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



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

# CLARA (Clustering Large Applications)

---

- ❑ *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)

---

- ❑ *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)