Data Mining: Concepts and Techniques

modified by: Manjunath 04/2003

Intelligent Database Systems Research Lab
School of Computing Science
Simon Fraser University, Canada
http://www.cs.sfu.ca

Chapter 8. Cluster Analysis

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 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

Types of Data/ Data Structures

- Data matrix

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

- Dissimilarity matrix

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

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.
- 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.

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

- Introduction
- 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

Score functions for partition-based clustering

- $d(x,y)$: distance between points x and y.
  - Clusters should be compact (*wc:* within cluster variation)
  - Clusters should be as far apart from each other as possible (*bc* :between cluster varitation)

Let $r_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$

n_k : number of points in the k-th cluster

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{r}_k)^2$$

Score function: examples

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{r}_k)^2$$ (spherical clusters)

$$bc(C) = \sum_{1 \le j < k \le K} d(\mathbf{r}_j, \mathbf{r}_k)^2$$

$$\text{score function} = \frac{bc(C)}{wc(C)}$$

Another example of *WC*: consider the distance to the nearest point in the same cluster (also referred to as the minimum distance or single link criterion –results in elongated clusters)

$$wc(C_k) = \max_i \min_{\mathbf{y}(j) \in C_k} \{ d(\mathbf{x}(i), \mathbf{y}(j)) \mid \mathbf{x}(i) \in C_k, \mathbf{x} \ne \mathbf{y} \}$$

The *K-Means* Clustering Method

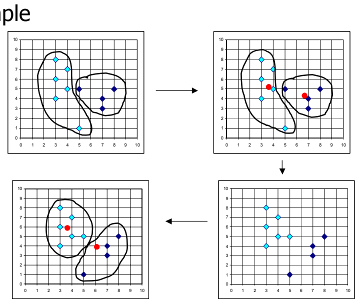
- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each object to the cluster with the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

Task: Given $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$; find K clusters $\{C_1, \ldots, C_K\}$

for $k = 1, .., K$ let $\mathbf{r}_k$ be a randomly chosen point from D;
while changes in clusters C_k happen do
  form clusters:
  for $k = 1, \ldots K$ do
    $C_k = \{\mathbf{x} \in D \mid d(\mathbf{r}_k, \mathbf{x}) \le d(\mathbf{r}_j, \mathbf{x}), \forall j \ne k\}$;
  end;
  compute new cluster centers:
  for $k = 1, \ldots, K$ do
    r_k = the vector mean of the points in C_k
  end;
end;

The *K-Means* Clustering Method

- Example



Comments on the *K-Means* Method

- Strength
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *K* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - 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'97)
  - Replacing means of clusters with <u>modes</u>
  - Using new dissimilarity measures to deal with categorical objects
  - Using a <u>frequency</u>-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

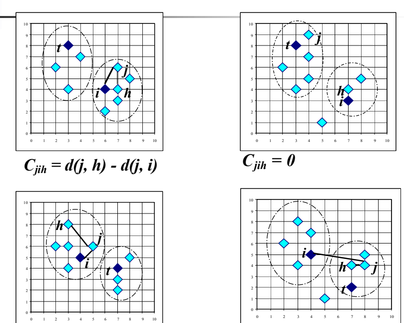
The *K-Medoids* Clustering Method

- Find *representative* objects, called <u>medoids</u>, 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
- Focusing + spatial data structure (Ester et al., 1995)

PAM (Partitioning Around Medoids) (1987)

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

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



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

$C_{jih} = 0$

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

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

CLARA (Clustering Large Applications) (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)*

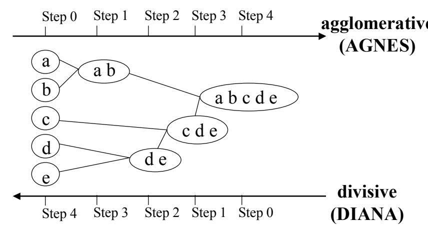
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* 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)

Chapter 8. Cluster Analysis

- Introduction
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- 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

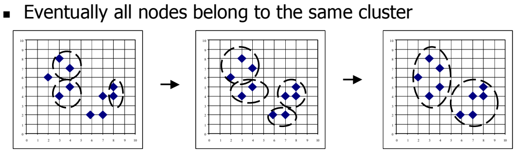


Agglomerative clustering

For *i = 1,…, n* let Ci = {x(i)}
While there us more than one cluster left **do**
 let Ci and Cj be the clusters minimizing the
 distance D(Ck, Ch) between any two clusters
 Ci = Ci U Cj;
 remove cluser Cj;
end

Eg: AGNES (Agglomerative Nesting)

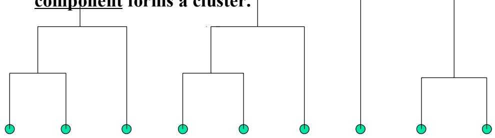
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

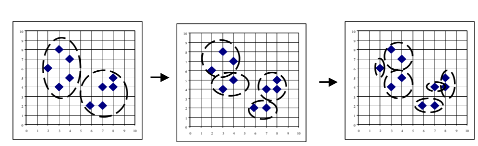
Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.

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



Hierarchical Clustering: +s

- Do not need to know vector representation of two objects as long as we can compute the distance between them (unlike partition based methods)
  - Eg. Clustering of protein sequences where well defined notions of distances between two sequences exist; --eg. Edit distance

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

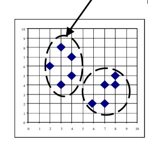
Clustering Feature: $CF = (N, \vec{LS}, SS)$

N: **Number of data points**

$LS: \sum_{i=1}^{N} \vec{X_i}$

$SS: \sum_{i=1}^{N} \vec{X_i}^2$

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

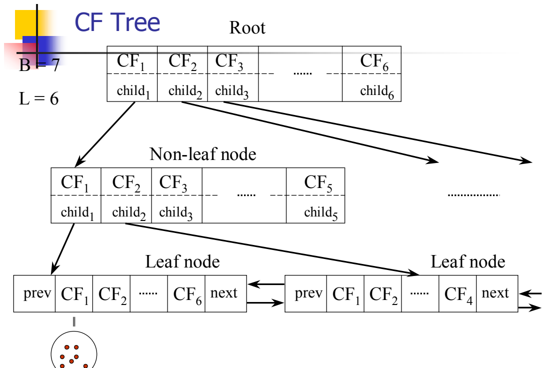


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF Tree

- A CF Tree is a height balanced tree with two parameters: branching factor B and threshold T.
- Each nonleaf node contains at most B entries of the form [CF(i), Child (i)] where child(i) is a pointer to the i-th child node. CF(i) is the CF of the subcluster represented by this child.
- A leaf node contains at most L entries
- All entries at a leaf node must satisfy a threshold requirement (average distance between points represented by that node or the diameter of the cluster)

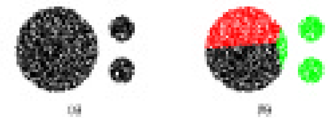
CF Tree



Birch Tree

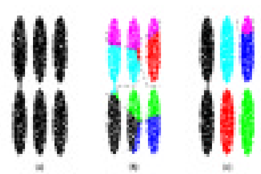
- Phase 1: tree is built dynamically as objects are inserted. An object is inserted into the closest leaf entry.
  - If the diameter of the subcluster after insertion exceeds a threshold T, then the leaf node (and possible the other nodes) are split.
  - After insertion, information is passed towards the root
- If the size of the CF tree exceeds the main memory constraint, the tree is rebuilt (with a new threshold)

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



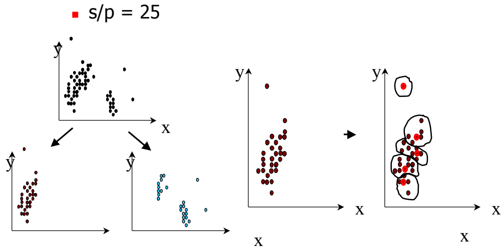
- Drawbacks of square-error based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex shaped, similar size and density, and if *k* can be reasonably estimated

Cure: The Algorithm

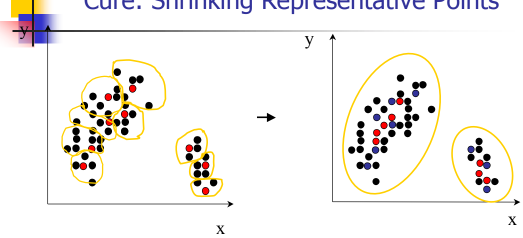
- Draw random sample *s*.
- Partition sample to *p* partitions with size *s/p*
- Partially cluster partitions into *s/pq* clusters
- Eliminate outliers
  - By random sampling
  - If a cluster grows too slow, eliminate it.
- Cluster partial clusters.
- Label data in disk

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

Jaccard coefficient

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

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

Rock: Algorithm

- Links: The number of common neighbours for the two points. E.g., at least two common elements

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

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

- Algorithm
  - Draw random sample
  - Cluster with links
  - Label data in disk