CLUSTER ANALYSIS

7. CLUSTER ANALYSIS

7.1 *What Is Cluster Analysis*

Cluster is a collection of data objects. Similar to one another within the same cluster. Dissimilar to the object in other clusters

Cluster Analysis

- Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- The process of grouping a set of abstract objects into classes of similar objects is called **clustering.**
- Clustering is an example of **unsupervised learning.**
- Clustering is a from of **learning by observation,** rather than **learning by examples.**

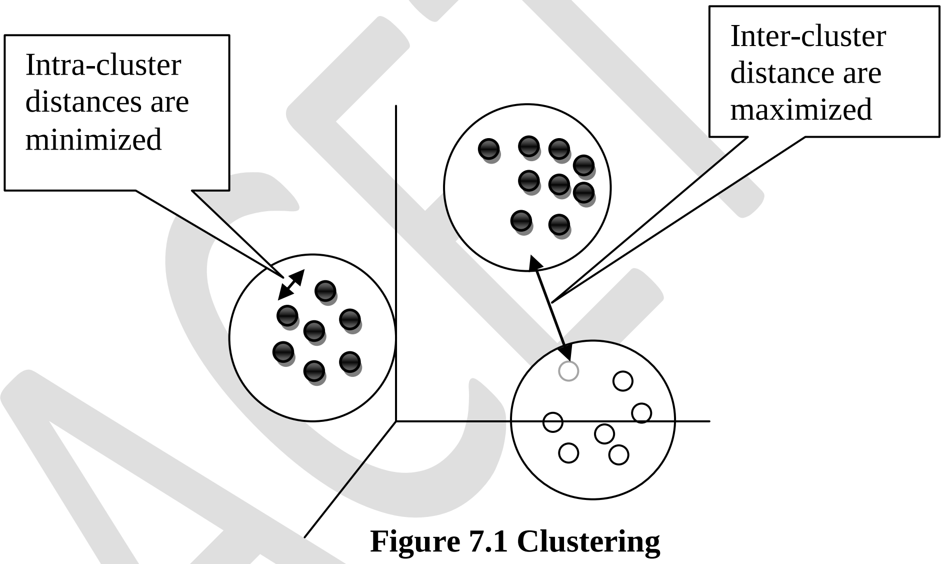

Intra-cluster distances are minimized
Inter-cluster distance are maximized

Figure 7.1 Clustering

Qualities of Good Clustering

- A good clustering method will produce high quality clusters with
    1. high intra-class similarity
    2. 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 the hidden patterns.

Measuring the Quality of Clustering

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

Desirable Properties of a Clustering Algorithm

- Scalability Pin terms of both time and space
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability and usability
- High dimensionality
- Ability to handle dynamic data

Cluster Analysis Applications

- Market research
- Pattern recognition
- Data analysis
- Image processing
- Web for information discovery

7.2 TYPE OF DATA IN CLUSTER ANALYSIS

Main memory - based clustering algorithms uses following data structures:

Data matrix (or *object-by-variable structure):* This represents *n* objects, such as persons, with *p* variables (also called *measurements or attributes*), such as age, height, weight, gender, and so on. The structure is in the from of a relational table, or *n*-by-*p* matrix (n objects ×p variables):

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

Dissimilarity matrix (or *object-by-object structure)*: This stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table:

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

Where d(i,j) is the measured difference or dissimilarity between objects I and j.

The rows and columns of the data matrix represent different entities, while those of the dissimilarity matrix represent the same entity. Thus, the data matrix is often called a two-mode matrix, where as the dissimilarity matrix is called a one-mode matrix. Many clustering algorithms operate on a dissimilarity matrix. If the data are presented in the from of a data matrix, it can first be transformed into a dissimilarity matrix before applying such clustering algorithms.

7.2.1 *Interval-Scaled Variables*

Interval-Scaled Variables describe measures that are commonly used for computing the dissimilarity of objects described by such variables. These measures include the *Euclidean, manhattan, and Minkowski distances.*

Standardize Data:

- Calculate the mean absolute deviation

$$S_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots |x_{nf} - m_f|\right)$$

Where $m_f = (x_{1f} + x_{2f} + \ldots\ldots + x_{nf})/n$

- Calculate the Z-score

$$Z_{if} = \frac{x_{if} - m_f}{S_f}$$

- Using mean absolute deviation is more robust than standard deviation. MAD is even more robust, but outliers disappear completely.

Given the Following measurement for the variable age:

18,22,25,42,28,43,33,35,56,28

Standardize the variable by the fallowing

1. Computer the mean absolute deviation of age
2. Computer the Z-score for the first four measurements.

(a)**Mean Absolute Deviation, S_f:**

$$S_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots |x_{nf} - m_f|\right)$$

Where $x_{if}, \ldots.., x_{nf}$ are measurements of f_x, m_f is the mean value of f.

$$m_f = \frac{330}{10} = 33$$

$S_f = \frac{1}{10}(|18 - 33| + |22 - 33| + |25\text{-}33|+|42\text{-}33|+|28\text{-}38|+|43\text{-}33|+|33\text{-}33|+|35\text{-}33|+|56\text{-}33|+|28\text{-}33|)$

$$= \frac{1}{10}(15 + 11 + 8 + 9 + 5 + 10 + 0 + 2 + 23 + 5)$$

$$= \frac{88}{10} = 8.8$$

b) Z-score (Standardized Measurement):

$$m_f = 33 \quad S_f = 8.8 \qquad Z_{if} = \frac{x_{if-m_f}}{S_f}$$

X_1=18

$$Z = \frac{18 - 33}{8.8} = -1.7$$

X=22

$$Z = \frac{22 - 33}{8.8} = -1.25$$

X=25

$$Z = \frac{25 - 33}{8.8} = -0.91$$

X=42

$$Z = \frac{42 - 33}{8.8} = 1.02$$

Similarity and Dissimilarity

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular distances are based on **Minkowshi distance (or Lp norm)**

$$d(i,k) = [|x_{i1} - x_{k1}|^p + |x_{i2} - x_{k2}|^p + \cdots \ldots + |x_{in} - x_{kn}|^p]^{\frac{1}{p}}$$

)-(**p=1: Manhattan distance (L_1 norm)**

$$d(i,k) = |x_{i1} - x_{k1}| + |x_{i2} - x_{k2}| + \ldots\ldots + |x_{in} - x_{kn}|$$

)-(**p=2: Euclidean distance (L_2 norm)**

$$d(i,k) = [|x_{i1} - x_{k1}|^2 + |x_{i2} - x_{k2}|^2 + \cdots \ldots + |x_{in} - x_{kn}|^2]^{\frac{1}{2}}$$

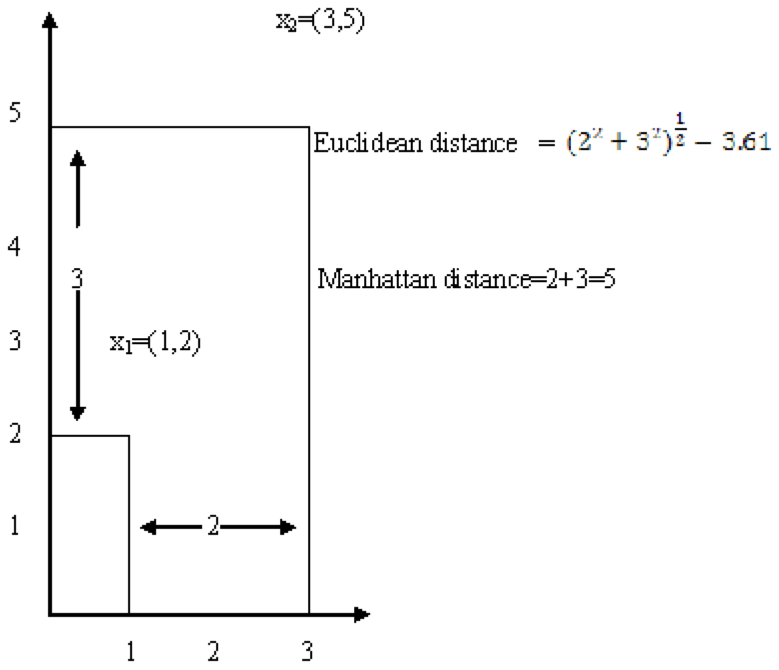


Figure 7.2

Properties of a distance

- d(i,k)>=0
- d(i,k)=0
- d(i,k) = d(k,i)
- d(i,j)<= d(i,k)+ d(k,j)

Example 1: Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8);

A) Compute the Euclidean distance between two objects.

B) Compute the Manhattan distance between the two objects.

C) Compute the Minkowski distance between the two objects using q=3.

(a) Euclidean distance:

d(i,j) = $\sqrt{(22-20)^2+(1-0)^2+(42-36)^2+(10-8)^2}$
$=\sqrt{2^2+1^2+6^2+2^2}$
$=\sqrt{4+1+36+4}$ = 6.708

(b) Manhattan distance:

d=|22-20|+|1-0|+|42-36|+|10-8|

=2+1+6+2 =11

(c) Minkowski Distance:

$$d(i,j)=\sqrt[3]{(|22-20|^3+|1-0|^3+|42-36|^3+|10-8|^3)}$$

$$=\sqrt[3]{(2^3+1^3+6^3+2^3)}$$

$$=\sqrt[3]{(8\quad+1\quad+216+8)}$$

$$=\sqrt[3]{233}=\mathbf{6.15}$$

Example2:

Give 5-dimensional numeric samples A=(1,0,2,5,3) and B=(2,1,0,3,-1).Find Euclidian distance between points.

Euclidian distance $d=\sqrt{(2-1)^2+(1-0)^2+(0-2)^2+(3-5)^2+(-1-3)^2}$

$=\sqrt{1^2+1^2+(-2)^2+(-2)^2+(-4)^2}$

$=\sqrt{26}$

= 5.099

7.2.2 Binary Variables

- Contingency table for binary data

Object j

Object j

	1	0	Sum
1	**Q**	**r**	**q+r**
0	**S**	**t**	**s+t**
sum	**q+s**	**r+t**	**P**

Figure 7.3 A contingency table for binary variables

Binary Variables

- Distance measure for symmetric binary variables. **Symmetric binary dissimilarity**

$$d(i,j)=\frac{r+s}{q+r+s+t}$$

- Distance measure for asymmetric binary variables.
- **Asymmetric binary dissimilarity.** Negative matches dropped

$$d(i,j)=\frac{r+s}{q+r+s}$$

- Asymmetric binary similarity

$$sum(i,j) = \frac{q}{p+q+r+s} = 1 - d(i,j)$$

This is also called the Jaccard coefficient

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	Y	N	N	N	N
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

- Gender is symmetric attribute
- Remaining attributes are asymmetric binary

Let values Y and P set to 1, and the value N is set to 0. Suppose that the distance between two patients is based on the asymmetric attributes

$$d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(Jim, Mary) = \frac{1+2}{1+1+2} = 0.75$$

7.2.3 Nominal or Categorical Variable

A generalization of the binary in that it can take more than 2 states

Method 1: Simple matching

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

Where m: # of matches p: total # number of variables

Method 2: use a large number of binary variables

Create a new binary 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
  - replaced x_{if} by their rank, r_{if} 2(1,…,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 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 Aexp(βt) or Aexp(-βt**)**

Methods:

- Treat them as interval-scaled variables – not a good choice (it can distort data)
- Apply logarithm transformation y_{if} =log(x_{if}) treat them as continuous ordinal data and their ranks as interval-valued

7.2.4 Variables of Mixed Types

- A database may contain all the six types of variables
- Weighted formula to combine their effects

$$d(i,j) = \frac{\Sigma_f \delta_{ij}^{(f)} d_{if}^{(f)}}{\Sigma_f \delta_{if}^{(f)}}$$

Where

$$\delta_{if}^{(f)} = \begin{cases} 0, x_{if,}\ x_{if} \text{ any missing or } x_{if} = x_{if} \\ 1, \text{otherwise} \end{cases}$$

And the contribution of f to dissimilarity $d_{if}^{(f)}$ is computed depending on its type

Computation of $d_{if}^{(f)}$

- If f is interval-based:

$$d_{if}^{(f)} = \frac{|x_{if} - x_{if}|}{max_h - min_h\ x_{hf}}$$

Where h runs over all non missing objects for variables f

- If f is binary or categorical:

$$d_{if}^{(f)} = \begin{cases} 0, x_{if} = x_{if} \\ 1, \text{otherwise} \end{cases}$$

- If f is ordinal compute ranks r_{if} and

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

And treat Z_{if} as interval-scaled

7.2.5 Vector Objects

- Vector objects: keywords in documents, gene features in micro-arrays, etc.
- Broad applications: information retrieval, biology taxonomy, etc.
- **Cosine measure**

$$s(x,y) = \frac{x'y}{||x|| - ||y||}$$

- **Tanimoto coefficient**

$$s(x,y) = \frac{x'y}{x'x + y'y - x'y}$$

7.3 A CATEGORIZATION OF MAJOR CLUSTERING METHODS

Partitioning Methods:

- Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k< =n.

- It classifies the data into k groups, which together satisfy the following requirements:

  1. Each group must contain at least one object.

  2. Each object must belong to exactly one group.

- The general criterion of a good partitioning is that object in the same cluster are “close” or related to each other, where object of different clusters are “far apart” or very different.

Example: K-means, K- Mediods

1)Hierarchical Methods

- A hierarchical method creates a hierarchical decomposition of the given set of data object i.e., the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object.

Two Main Types of Hierarchical Clustering:

Agglomerative: (***Bottom-up Approach)***

- Start with the points as individual clusters

- At each step, merge the cluster until only one cluster (or k clusters) left

Division: (***Top-Down Approach)***

- Start with one, all-inclusive cluster
- At each step, split a cluster contains a point (or there are k clusters)

To improving the quality of hierarchical clustering:

1. Perform careful analysis of object "linkages" at each hierarchical partitioning.

2. Integer hierarchical agglomeration and other approaches by first using a hierarchical agglomerative algorithm to group objects into micro clusters, and then performing macroclustering on the microclusters using another clustering methods.

Example: CURE, BIRCH, CHAMELEON

2) ***Density-Based Methods***

- Most partitioning methods cluster objects based on the distance between objects, such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes.
- Other clustering methods have been developed based on the notion of density.
- Their general idea is to continue growing the given cluster as long as the density (number of object or data points) in the "neighborhood" exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
- Such a method can be used to filter out noise (outliers) and discover clusters of arbitrary shape.

Example: DBSCAN, OPTICS, DENCLUE

3) Grid-Based Methods

- Grid –based methods quantized the space into a finite number of cells that from a grid structure.
- All of the clustering operations are performed on the grid structure (i.e., on the quantized space).
- The main advantage of this approach is its fast processing time, which is typically independent of the number of data object and dependent only on the number of cells in each dimension in the quantized space.

Example: STING, WaveCluster, CLIQUE

4) Model-Based Methods:

- Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model.
- A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.
- It also leads to a way of automatically determining the number of clusters based on standard statistics, taking “noise” or outliers into account and thus yielding robust clustering methods.

Example: COBWEB, CLASSIT

7.4 PARTITIONING METHODS

Partitioning algorithms construct partitions of a data base of N objects into a set of k clusters. The construction involves determining the optimal partition with respect to an objective function. In general, partitioning algorithms is

- Nonhierarchical, each instance is placed in exactly one of K nonoverlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K.

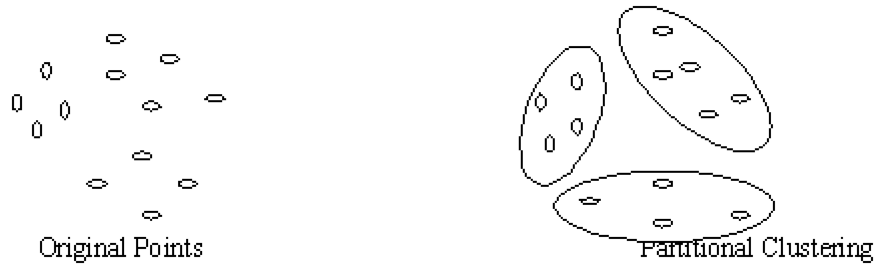


Figure 7.4 Partitional Clustering

- The Partitional techniques usually produce clusters by optimizing a criterion function defined either locally or globally. A global criterion, such as the Euclidean square-error measure, represents each cluster by a prototype or centroid and assigns the samples to clusters according to the most similar prototypes. A local criterion, such as the minimal mutual neighbor distance (MND0, forms cluster by utilizing the local structure or context in the data.

The most commonly used Partitional-clustering strategy is based on the square error criterion. The general objective is to obtain the partition that, for a fixed number of clusters, minimizes the total square error. Suppose that the given dataset of N samples in an n-dimensional space has been partitioned into k-clusters$\{c_1 , c_2 ,\ldots, c_k\}$.

Each c_k has n_k samples and each sample has exactly one cluster, so that

$\sum n_k = N$ where k=1, …., k.

The mean vector m_k cluster C_k is defined as the centroid of the cluster

$$M_k = (1/n_k) \sum_{i=1}^{n_k} X_{ik}$$

Where X_{ik} is the sample belonging to cluster C_k.

The square-error for cluster C_k is the sum of the squared Euclidean distance between each sample in C_k and its centroid. This error is also called the within-cluster variation.

$$e_k^2 = \sum_{i=1}^{n_k} (X_{ik} - M_k)^2$$

The square-error for the entire clustering space containing k clusters is the sum of the within-cluster variations:

$$E_k^2 = \sum_{k=1}^{k} e_k^2$$

7.4.1 Classical partitioning Methods: k-Means and k-Medoids

The most well-known and commonly used partitioning methods are k-means, k-Medoids, and their variations.

Centroid-Based Technique: The k-Means Method

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. Cluster similarity is measured in regard to the mean value of the object in a cluster, which can be viewed as the cluster's centroid or center of gravity.

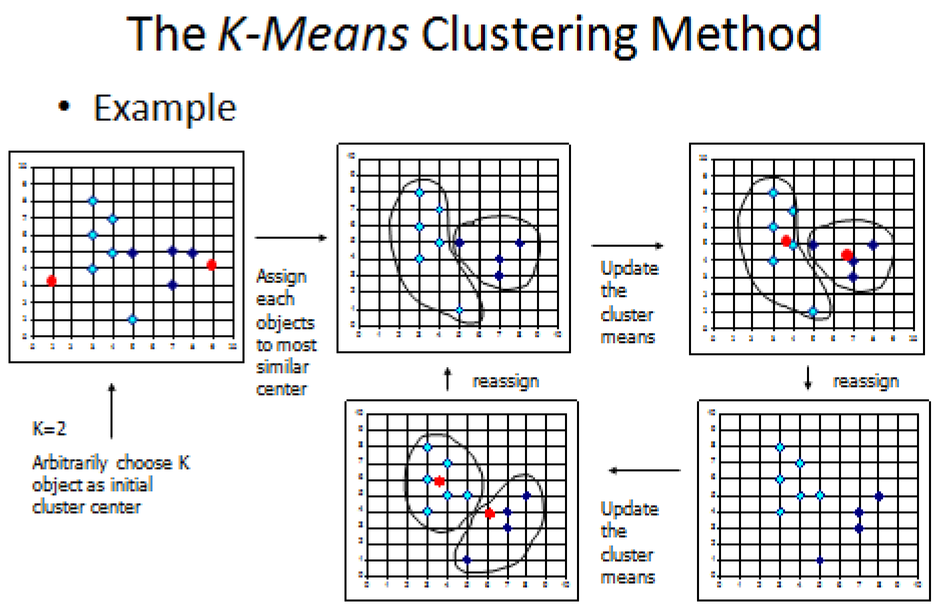
Given k, the k-means algorithm is implemented in four steps:

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, i.e., 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

Finally, this algorithm aims at minimizing an objective function, in this case a mean squared error function is calculated as:

$$E = \sum_{i=1}^{k} \sum_{p \in c_i} |p - m_i|^2$$

Where E is the sum of the square error for all objects in the data set; p is the point space representing a given objects; and m_i is the mean of cluster C_i (both p and m_i are multidimensional).


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

Figure 7.5 K-Means Clustering

Algorithm:

k-means. The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input

K: the number of clusters,

D: a data set containing n objects.

Output: A set of k clusters

Methods:

1. Select initial partitions with k clusters containing randomly chosen samples, and compute the centroids of the clusters.
2. Generate a new partition by assigning each sample to the closest cluster center.
3. Compute new cluster centers as the centroids of the cluster.
4. Repeat step2 and 3 until an optimum value of the criterion function is found or until the cluster membership stabilizes.

Scaling the k-means Algorithm

- Identify three kinds of regions in data: regions that are compressible, regions that must be maintained in main memory, and regions that are discardable.
- If an object is neither discardable nor compressible, then it should be retained in main memory.
- To achieve scalability, the iterative clustering algorithm only includes the clustering features of the compressible objects and the objects that must be retained in main memory, thereby turning a secondary-memory based algorithm into a main-memory based algorithm.
- Another approach is to perform microclustering, which first groups nearby objects into "microclusters" and then performs k-means clustering on the microclusters.

Advantages

1. With a large number of variables, k-means nay be computationally faster than hierarchical clustering (if k is small).
2. K-means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

Disadvantages

- Difficult in comparing the quality of the clusters produced.
- Applicable only when mean is defined.
- Need to specify k, the number of clusters, in advance.
- Unable to handle noisy data and Outliers.
- Not suitable to discover cluster with non-convex shapes.

Representative Object-Based Technique: The k-Medoids Method

This algorithm is very similar to k-Means with the small exception of instead of creating an artificial point, k-Medoids recalculates from the nearest actual point in a data set. The reason for this is that it is not acceptable to outliers that are extremely far away from it.

K-medoids clustering works as following:

- The initial representative objects (or seeds) are chosen arbitrarily.
- The iterative process of replacing representative objects by nonrepresentative objects continues as long as the quality of the resulting clustering is improved.
- Quality is estimated using a cost function that measures the average dissimilarity between an object and the representative object of its cluster.
- To determine whether a nonrepresentative object, o_{random}, is a good replacement for a current representative object, 0_j, the following four cases are examined for each of the nonrepresentative objects, p.

Case 1: p currently belongs to representative object, o_j. If o_j is replaced by o_{random} as a representative object and p is closest to one of the other representative objects, o_j, $i \neq j$, then p is reassigned to o_i.

Case 2: p currently belongs to representative object, o_i. if o_j is replaced by 0_{random} as a representative object and p is closest to o_{random}, then p is reassigned to o_{random}.

Case 3: p currently belongs to representative object, o_i. $i \neq j$ is replaced by 0_{random} as a representative object and p is still closest to o_i, then the assignment does not change.

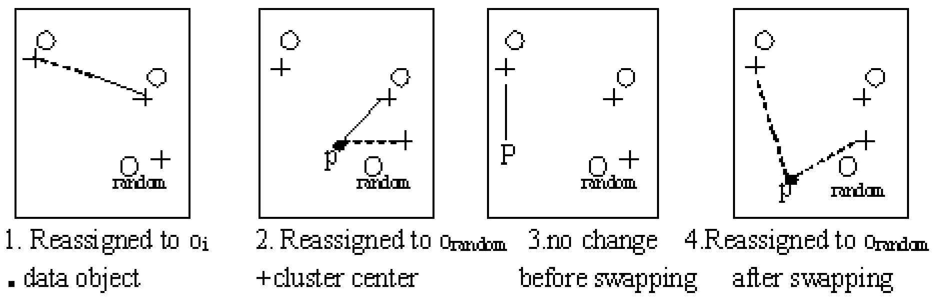


Figure 7.6 Four cases of the cost function for k-medoids clustering

Case 4: p currently belongs to representative object, o_i. $i \neq j$. if o_j is replaced by 0_{random} as a representative object and p is closest to o_{random}, then p is reassigned to o_{ramdom}.

PAM (Partitioning Around Medoids) was one of the first k-medoids algorithms introduced. It attempts to determine k partitions for n objects. After an initial random selection of k representative objects, the algorithm repeatedly tries to make a better choice of cluster representative. The final set of representative objects are the respective medoids of the clusters (or) centered objects.

Algorithm: k-medoids. PAM, a k-medoids algorithm for partitioning based on medoid clusters. Or central objects.
Input:
K: the number of clusters.
D: a data set containing n objects.
Output: A set of k clusters.
Method:

```
1. Arbitrarily choose k objects in D as the initial representative objects or seeds;
2. Repeat
3. Assign each remaining object to the cluster with the nearest representive
   object;
4. Randomly select a non representive object, $o_{random}$
5. Compute the total cost, s, of swapping representive object, $o_j$, with $o_{random}$;
6. If $s<0$ then swap $o_j$ with $o_{random}$ to form the new set of k represntive objects;
7. Until no change;
```

Advantages

- Pam is very robust to the existence of outliers. The clusters found by this method do not depend on the order in which the objects are examined.
- PAM works effectively for small data sets.

Disadvantages

- It does not sale well for large data sets.

7.4.2 Partitioning methods in large databases

From k-medoids to ClARANS

CLARA (clustering large applications) overcomes the problem of scalability

that k-medois suppers from. CLARA draws a simple of data set, and applies PAM on this sample to determine the optimal set of mediods from the sample. If then classifies the remaining objects using the partitioning principle. If the sample were drawn in a sufficiently random way, the methods of the sample would approximate the mediods of the entire data set.

CLARA algorithm:

```
Input: Database of d objects.
Repeat for m times
        Draw a sample s D randomly from D
        Call PAM(s,k) To get k mediods.
        Classify the entire data set D to $c_1$,$c_2$, -$c_k$.
        Caliculate the quality of clustering as the average dissimilarity
End
```

To improve the quality of CLARA, we go for CLARANS (clustering large applications based on RANdomized search). Instead of clustering on the whole dataset, CLARANS clusters on a sample of the database.

CLARANS works as follows:

- CLARANS draws sample with some randomness in each step of search
- At each step, PAM examines all the neighbors of the current node in its search for a minimum cost solution. The current node is then repulse by the neighbor with the largest descent in costs.
- If a better neighbor is found(i.e having a low error), CLARANS moves to the neighbors node and the process starts again;
- Otherwise, the current clustering produces a local minimum. If a local minimum is found, CLARANS starts with new randomly selected nodes in search of a new local minimum.
- Once a user-specified number of a local minima has been found, the algorithm output, as a solution, the best local minimum, that is, the local minimum having the lowest cost.

Advantages

- More efficient then k-means and k-mediods.
- Used to detect outliers.

Dis advantages

- Me not a find a local minimum ue to the trimming of its searching.
- It assumes that all objects fit into the main memory, and the result is sensitive to input order.

7.5 HIERARCHICAL METHOD

Basically hierarchical methods group data into a tree of clusters. There are two basic varieties herarsial algorithms; agglomerative and divisive. A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering.

7.5.1 Agglomerative and divisive hierarchical clustering

- **Agglomerative hierarchical methods (bottom-up strategy)**
  Begin with as many clusters as objects. Clusters are successively merged until only one cluster remain.
- **Divisive hierarchical methods(top-down strategy)**
  Begin with all objects in one cluster. Group are continually divided until there are as many clusters as objects.

Steps in agglomerative hierarchical clustering

1. Compute the proximity matrix.
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters.
5. Update the proximity matrix
6. **Until** only a single cluster remains

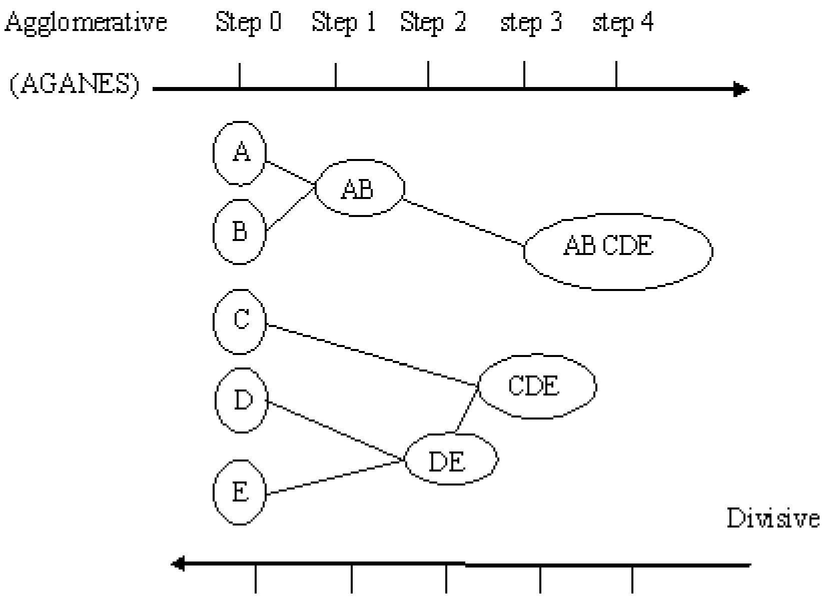


Figure 7.7 agglomerative and divisive hierarchical clustering on data objects{a,b,c,d,e}

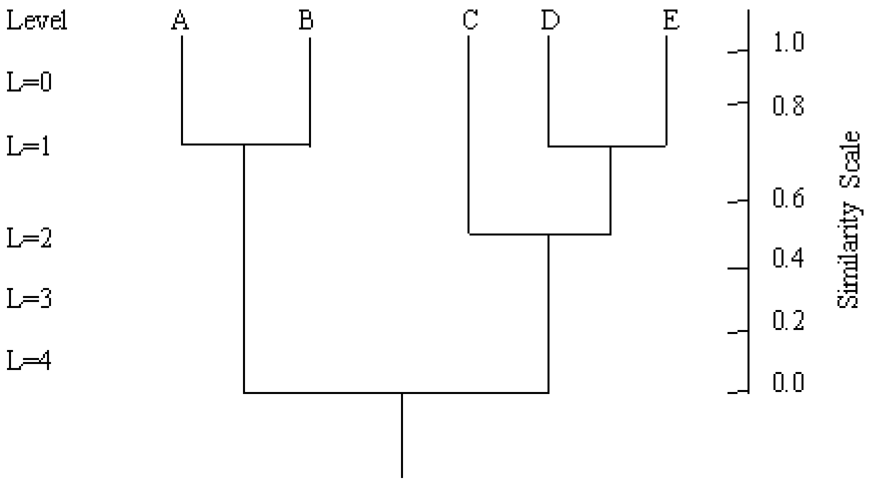


Figure 7.8dendrogram representation for hierardhical clustering of data objects {a,b,c,d,e}

7.5.2 BIRCH: Balanced Iterative Reducing and Clustering Using Hierarchies

BIRCH partitions objects hierarchically using tree structures and then refines the clusters using other clustering methods. I t defines clustering feature and an associated tree structure that summarizes a cluster. The tree (CF tree) is a height-balanced tree that stores cluster information. BIRCH doesn't produce spherical clusters and may produce unintended cluster. These structures help the clustering method achieve good speed and scalability in large databases and also make it effective for incremental and dynamic clustering of incoming objects.

BIRCH applies a *multiphase* clustering technique, a single scan of the data set yields a basic good clustering, and one or more additional scans can (optionally) be used to further improve the quality.
The primary phases are:

Phase 1: BIRCH scans the database to build an initial in-memory CF tree, which can be viewed as a multilevel compression of the data that tries to preserve the inherent clustering structure of the data.
Phase 2: BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.

- **Clustering features**
  - Summary of the statistics for a given subclusters, the 0th, 1st, and 2nd moments of the subcluster from the statistical point of view.
  - Registers crucial measurements for computing cluster and utilizes storage efficiently.

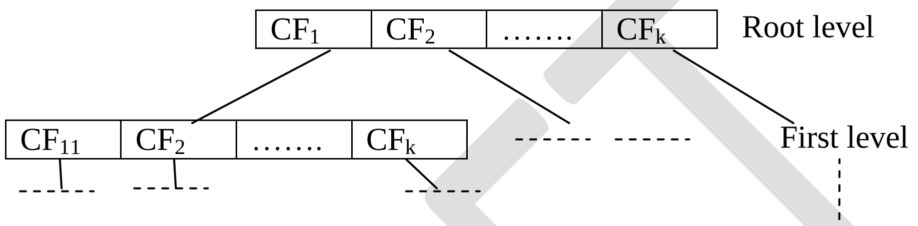


Figure 7.9 CF Tree

- A CF tree is a height balanced tree that stores the clustering features for a hierarchical clustering.
  - A non leaf node in a tree has descendents or "Children".
  - The non leaf nodes stores sums of the CFs of their children.
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children.
  - Threshold : Maximum diameter of sub-clusters stored at the left nodes

Drawback

Handles only numeric data and sensitive to the order of the data record.

7.5.3 CURE: Clustering Using Representatives

Creates cluster by sampling the database and shrinks them toward the center of the cluster by a specified fraction. Obviously better in runtime but lacking in precision. It is robust against outliers and creates clusters of differing sizes that are not necessarily spherical. CURE fails when clustering categorical data. Ignores aggregate interconnectivity of objects in separate clusters.

Steps:

- Draw a random sample, of the original objects.
- Partition sample S into a set of partitions and form a cluster for each partion.
- Representative points are found by selecting a constant number of points from a cluster and then "shrinking" them toward the centre of the cluster.

- Cluster similarity is the similarity of the closest pair of representative points from different clusters
- Shrinking representative points toward the center helps avoid problems with noise and outliers
- CURE is better able to handle clusters of arbitrary shapes and sizes

7.5.4 ROCK (Robust Clustering using links) : A Hierarchical Clustering Algorithm for Categorical Attributes

Merges clusters based on their interconnectivity. Great for categorical data. Ignores information about the looseness of two clusters while emphasizing interconnectivity.

Steps:

- Obtain a sample of points from the data set
- Compute the link value for each set of points, i.e., transform the original similarities (computed by Jaccard coefficient) into similarities that reflect the number of shared neighbors between points
- Perform an agglomerative hierarchical clustering on the data using the "number of shared neighbors" as similarity measure and maximizing "the shared neighbors" objective function
- Assign the remaining points to the clusters that have been found

Link Measure in Rock

Links : Number of common neighbors

- C_1 < a, b, c, d, e> : {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {c, d, e}
- C_2 <a, b, f, g> : {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}

  Let T1= {a, b, c}, T_2={c, d, e}, T_3= {a, b, f}
- Link (T_1, T_2) = 4, since they have 4 common neighbors. i.e., {a, c, d}, {a, c, e}, {b, c, d}, {b, c, e}
- Link (T_1,T_2) = 3, since they have 3 common neighbors. i.e., {a, b, d}, {a, b, e}, {a, b, g}

7.5.5 Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling

Uses dynamic modeling. It basically constructs a sparse k-nearest neighbor graph, then partitions the graph into pieces and then clusters the pieces together. Produces more natural clusters than DBSCAN (It is possible to have non-spherical clusters.) Uses density measurements to determine the k-nearest neighbor.

Features:

- Adapt to the characteristics of the data set to find the natural clusters
- Use a dynamic model to measure the similarity between clusters
  - ❖ Main property is the relative closeness and relative inter-connectivity of the cluster

- ❖ Two clusters are combined if the resulting cluster shares certain *properties* with the constituent clusters
- ❖ The merging scheme preserves *self-similarity*

- One of the areas of application is spatial data

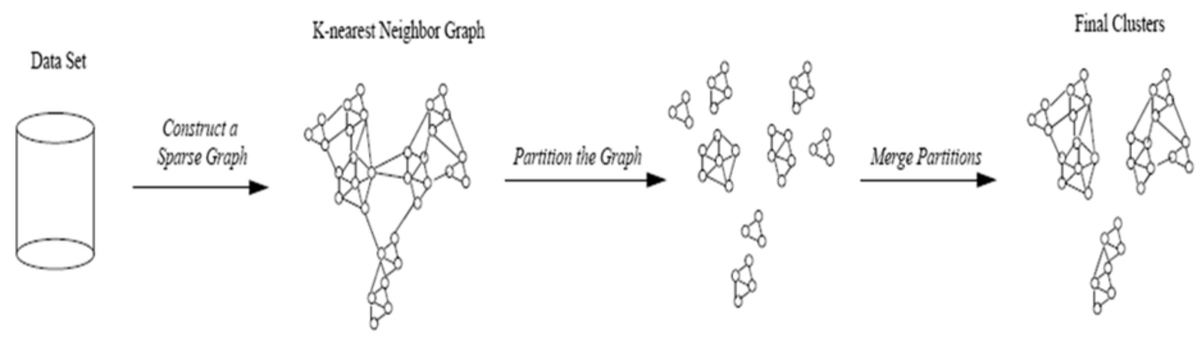


Figure 7.10 Chameleon: Hierarchical clustering based on k-nearst neighbors and dynamic
modeling.

Chameleon Steps

- Preprocessing – Represent the Data by a Graph
  - ❖ Given a set of points, construct the k-nearest-neighbor (k-NN) graph to capture the relationship between a point and its k nearest neighbors
  - ❖ Concept of neighborhood is captured dynamically (even if region is sparse)
- **Phase 1:** Use a multilevel graph partitioning algorithm on the graph to find a large number of clusters of well-connected vertices
  - – Each cluster should contain mostly points from one “true” cluster, i.e., is a sub-cluster of a “real” cluster
- **Phase 2:** Use Hierarchical Agglomerative Clustering to merge sub-clusters
  - – Two clusters are combined if the *resulting cluster shares certain properties with the constituent clusters*
  - – Two key properties used to model cluster similarity:
    - ➢ Relative Interconnectivity: Absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters
    - ➢ Relative Closeness: Absolute closeness of two clusters normalized by the internal closeness of the clusters