Partitioning Methods

Cluster Analysis:

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.
A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.

A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.

Cluster analysis tools based on k-means, k-medoids, and several methods have also been built into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.

Partitioning Methods:

A partitioning method constructs *k* partitions of the data, where each partition represents a cluster and $k <= n$. That is, it classifies the data into *k* groups, which together satisfy the following requirements:

Each group must contain at least one object, and Each object must belong to exactly one group. A partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are close or related to each other, whereas objects of different clusters are far apart or very different.

The most well-known and commonly used partitioning methods are

- The *k*-Means Method
- k-Medoids Method

K-Means Method:

The k-means algorithm takes the input parameter, k, and partitions a set of n objects intok 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 objects in a cluster, which can be viewed as the cluster's centroid or center of gravity. The *k*-means algorithm proceeds as follows.

- First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center.
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- It then computes the new mean for each cluster.
- This process iterates until the criterion function converges.
- Typically, the square-error criterion is used, defined 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 in space representing a given object mi is the mean of cluster Ci

k-means partitioning algorithm:

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

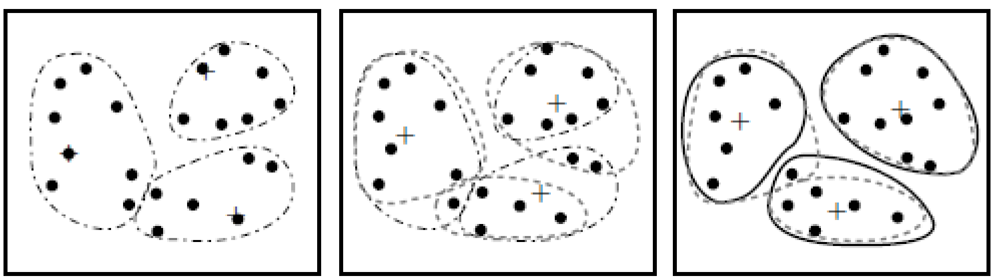
Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
(2) **repeat**
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;



Clustering of a set of objects based on the *k*-means method

k-Medoids Method:

- The k-means algorithm is sensitive to outliers because an object with an extremely large value may substantially distort the distribution of data. This effect is particularly exacerbated due to the use of the square-error function.
- Instead of taking the mean value of the objects in a cluster as a reference point, we can pick actual objects to represent the clusters, using one representative object per cluster. Each remaining object is clustered with the representative object to which it is the most similar.

- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. That is, an absolute-error criterion is used, defined as

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|,$$

Where *E* is the sum of the absolute error for all objects in the data set ***p*** is the point in space representing a given object in cluster *Cj* **oj** is the representative object of Cj

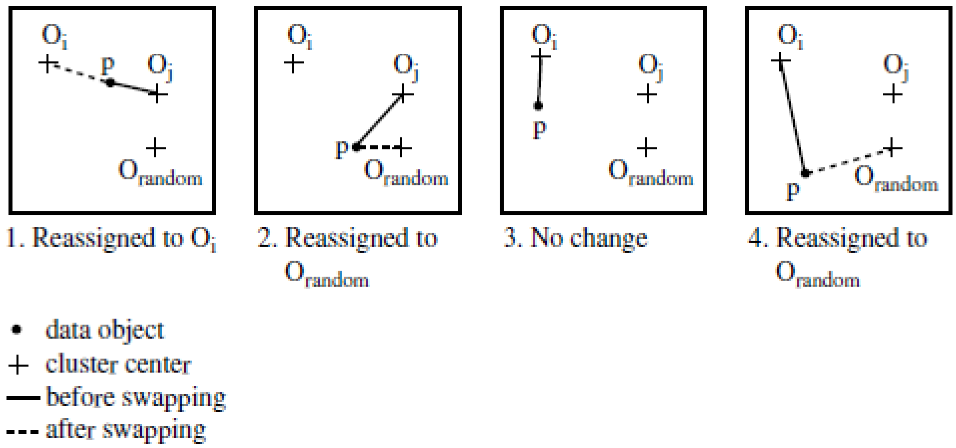
- The initial representative objects are chosen arbitrarily. The iterative process of replacing representative objects by non representative objects continues as long as the quality of the resulting clustering is improved.
- This quality is estimated using a cost function that measures the average dissimilarity between an object and the representative object of its cluster.
- To determine whether a non representative object, oj random, is a good replacement for a current representative object, oj, the following four cases are examined for each of the non representative objects.

Case 1: p currently belongs to representative object, oj. If ojis replaced by orandom as a representative object and p is closest to one of the other representative objects, oi,i≠j, then p is reassigned to oi

Case 2: p currently belongs to representative object, oj. If ojis replaced by orandom as a representative object and p is closest to orandom, then p is reassigned to orandom.

Case 3: p currently belongs to representative object, oi, i≠j. If ojis replaced by orandomas a representative object and p is still closest to oi, then the assignment does notchange.

Case 4: p currently belongs to representative object, oi, i≠j. If ojis replaced by orandom as a representative object and p is closest to orandom, then p is reassigned to orandom.



Four cases of the cost function for *k*-medoids clustering

***k*-MedoidsAlgorithm:**

The k-medoids algorithm for partitioning based on medoid 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 representative object;
(4) randomly select a nonrepresentative object, o_{random};
(5) compute the total cost, S, of swapping representative object, o_j, with o_{random};
(6) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of k representative objects;
(7) **until** no change;

The *k*-medoids method ismore robust than *k*-means in the presence of noise and outliers, because a medoid is lessinfluenced by outliers or other extreme values than a mean. However, its processing ismore costly than the *k*-means method.