k-Medoids: A Representative Object-Based Technique.

- 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~~ representative object per cluster.
- Each remaining object is assigned to the cluster of which the representative object is the most similar
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each point 'p' & its corresponding representative object.
- That is, an absolute-error criterion is used,

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i) \qquad \text{where,}$$

E is the sum of the absolute error for all objects p in the data set.

o_i is the representative object of C_i.

when $k=1$, we can find the exact median in $O(n^2)$ time.

k-medoids clustering methods

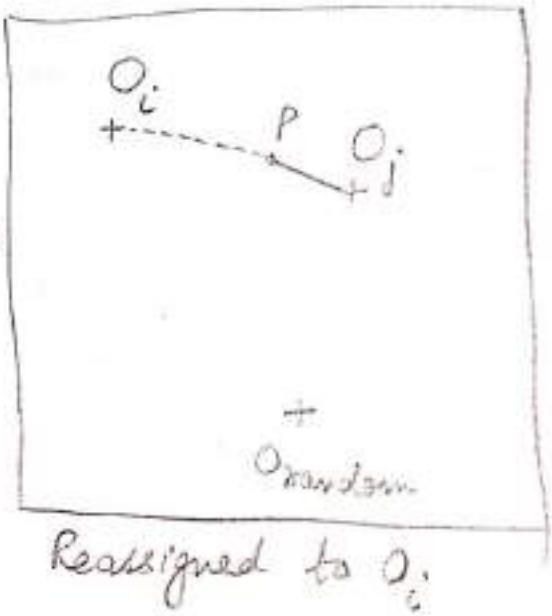
1. Partitioning Around Medoids (PAM)
2. Clustering LARge Applications (CLARA)

Partition Around Medoids (PAM)

- PAM is a realization of k-medoids clustering.
- It tackles the problem in an iterative, greedy way.
- Like k-means algorithm, the initial representative objects are chosen arbitrarily.
- Next, we consider whether replacing a representative object by a non-representative object would improve the clustering quality.
  - All possible replacements are carried out.
  - The process continues until the quality of the resulting clustering cannot be improved by any replacement.
  - The quality is measured by a cost f" of the average dissimilarity b/w an object & the representative object of its cluster.
- Let $o_1, o_2, o_3, \ldots o_k$ be the current set of representative objects (ie. medoids)
- To determine whether a non-representative object, denoted by o_{random} is a good replacement for a current medoid o_j $(1 \le j \le k)$, we calculate the distance from every object p to the closest object in the set $\{o_1, o_2, \ldots o_{j-1}, o_{random}, o_{j+1}, \ldots o_k\}$, & use the distance to update the cost function.

- The strategy then iteratively replaces one of the medoids by one of the non-medoids as long as the quality of the resulting clustering is improved.
- To determine whether a nonmedoid object, O_{random}, is a good replacement for a current medoid, O_j, the following four cases are examined for each of the nonmedoids objects, p:

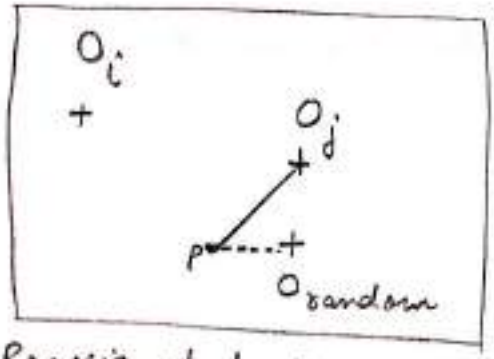
Case I:- p currently belongs to medoid O_j. If O_j is replaced by O_{random}. O_j ~~is replaced by O_{random}~~ as a medoid & p is closest to one of O_i, $i \neq j$, then p is reassigned to O_i.



Reassigned to O_i

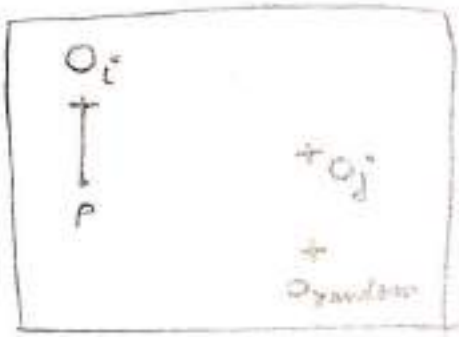
p · data object
+ cluster center
— before swapping
···· after swapping.

Case II :- 'p' currently belongs to medoid o_j. If o_j is replaced by o_{random} as a medoid & p is closest to o_{random}, the p is reassigned to o_{random}.



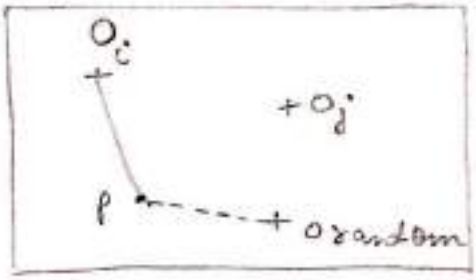
Reassigned to o_{random}

Case III : 'p' currently belongs to medoid o_i, $i \neq j$. If o_j is replaced by o_{random} as a medoid & p is still closest to o_i, then the assignment does not change.



No change

Case IV. 'p' currently belongs to medoid o_i. $i \neq j$. If o_j is replaced by o_{random} as a medoid & 'p' is closest to o_{random}, then 'p' is reassigned to o_{random}.



- Each time a reassignment occurs, a difference in absolute error, E, is contributed to the cost function.
- ∴ the cost f^n calculates the difference in absolute-error value if a current representative object is replaced by a non-representative object.
- The total cost of swapping (replacement) is the sum of costs incurred by all non representative objects.
- If the total cost is 'negative', then o_j is replaced or swapped with o_{random} because actual absolute error 'E' is reduced.
- If the total cost is 'positive', the current representative object o_j is considered acceptable & nothing is changed in the iteration.

Algorithm: k-medoids :- PAM :- a k-medoid algorithm for partitioning based on medoid or central ~~medoid~~ objects.

Input :- k : The no. of clusters., D :- dataset containing n objects

output :- set of k clusters

Method :-

(1) arbitrarily choose k-objects in D as initial rep. objects or seeds

(2) repeat

(3) assign each remaining object to the cluster with the nearest representative object.

(4) randomly select a non-representative 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;

Comparison between k-means & k-medoids:-

1) k-medoid method is more robust than k-means in the presence of noise & outliers because a medoid is less influenced by outliers or other extreme values than a mean.

2) Complexity of each iteration in the k-medoid algorithm is $O(k(n-k)^2)$. For large d/B (where n & k is very high), such computations become very costly & more costly than the k-means methods.

3) Both methods require the user to specify k, the number of clusters.

How to scale up k-medoids methods?

disadv :- PAM (A k-medoid) Algorithm work well for small databases & but not for large data sets. To deal with larger data sets, a sampling-based method CLARA (Clustering LARge Applications) can be used.

CLARA

- Instead of taking the whole dataset into consideration, CLARA uses a random sample of the data set.
- The PAM algorithm is then applied to compute the best medoids from the sample.
- The sample should closely represent the original data set.
- In many cases, a large sample works well if it is created so that each object has equal probability of being selected into the sample.
- CLARA builds clustering from multiple random samples & returns the best clustering as the output.

The complexity of computing the medoids on a random sample is $O(ks^2 + k(n-k))$, where s is the size of sample, k is no. of clusters & n is the total no. of objects.

- The effectiveness of CLARA depends on the sample size.

disadv:- If an object is one of the best k-medoids but is not selected during sampling, CLARA will never find the best clustering.

sol^n:- To improve the quality & scalability of CLARA, a randomized Algorithm called CLARANS (Clustering Large Applications based upon RANdomized Search) presents a trade-off b/w the cost & effectiveness of using samples to obtain clustering.

method:-
- first, it randomly selects k-objects in the data set as the current medoid.
- It then randomly selects a current medoid x & an object y that is not one of the current medoids.
- Replacement of 'x' by 'y' is done if it improves the absolute-error criterion.
- CLARANS conducts such randomized search l-times.
- The set of the current medoids after the 'l' steps is considered a local optimum.
- CLARANS repeat this randomized process 'm' times & returns the best local optimal as the final result.