https://www.cs.umb.edu/cs738/pam1.pdf

The PAM Clustering Algorithm

PAM stands for "partition around medoids". The algorithm is intended to find a sequence of objects called *medoids* that are centrally located in clusters. Objects that are tentatively defined as medoids are placed into a set S of *selected objects*. If O is the set of objects that the set $U = O - S$ is the set of *unselected objects*.

The goal of the algorithm is to minimize the average dissimilarity of objects to their closest selected object. Equivalently, we can minimize the sum of the dissimilarities between object and their closest selected object.

The algorithm has two phases:

(i) In the first phase, BUILD, a collection of k objects are selected for an initial set S.

(ii) In the second phase, SWAP, one tries to improve the quality of the clustering by exchanging selected objects with unselected objects.

The goal of the algorithm is to minimize the average dissimilarity of objects to their closest selected object. Equivalently, we can minimize the sum of the dissimilarities between object and their closest selected object.

For each object p we maintain two numbers:

- D_p, the dissimilarity between p and the closest object in S, and
- E_p, the dissimilarity between p and the second closest object in S.

These numbers *must be updated every time when the sets S and U change*. Note that $D_j \leq E_j$ and that we have $p \in S$ if and only if $D_p = 0$.

The BUILD phase entails the following steps:

1. Initialize S by adding to it an object for which the sum of the distances to all other objects is minimal.
2. Consider an object $i \in U$ as a candidate for inclusion into the set of selected objects.
3. For an object $j \in U - \{i\}$ compute D_j, the dissimilarity between j and the closest object in S.

4. If $D_j > d(i,j)$ object j will contribute to the decision to select object i (because the quality of the clustering may benefit); let $C_{ji} = \max\{D_j - d(j,i), 0\}$.

5. Compute the total gain obtained by adding i to S as $g_i = \sum_{j \in U} C_{ji}$.

6. Choose that object i that maximizes g_i; let $S := S \cup \{i\}$ and $U = U - \{i\}$.

These steps are performed until k objects have been selected. The decisions taken in assesing object i are shown in Figure 1.

The second phase, SWAP, attempts to improve the the set of selected objects and, therefore, to improve the quality of the clustering.

This is done by considering all pairs $(i,h) \in S \times U$ and consists of computing the effect T_{ih} on the sum of dissimilarities between objects and the closest selected object caused by swapping i and h, that is, by transferring i from S to U and transferring h to from U to S.

The computation of T_{ih} involves the computation of the contribution K_{jih} of each object $j \in U - \{h\}$ to the swap of i and h. Note that we have either $d(j,i) > D_j$ or $d(j,i) = D_j$.

1. K_{jih} is computed taking into account the following cases (also, see Figure 2):

   (a) if $d(j,i) > D_j$, then two subcases occur:
      i. if $d(j,h) \geq D_j$, then $K_{jih} = 0$;
      ii. if $d(j,h) < D_j$, then $K_{jih} = d(j,h) - D_j$.

   In both subcases, $K_{jih} = \min\{d(j,h) - D_j, 0\}$.

   (b) if $d(j,i) = D_j$, we have two subcases:
      i. if $d(j,h) < E_j$, where E_j is the dissimilarity between j and the second closest selected object, then $K_{jih} = d(j,h) - D_j$; note that K_{jih} can be either positive or negative.
      ii. if $d(j,h) \geq E_j$, then $K_{jih} = E_j - D_j$; in this case $K_{jih} > 0$.

   In each of the above subcases we have

   $$K_{jih} = \min\{d(j,h), E_j\} - D_j.$$

2. Compute the total result of the swap as

   $$T_{ih} = \sum \{K_{jih} \mid j \in U\}.$$

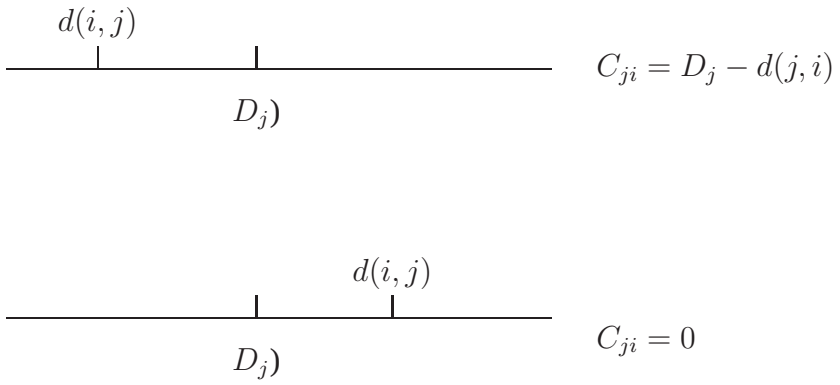

Figure 1: Computation of the Contribution C_{ji}

3. Select a pair $(i, h) \in S \times U$ that minimizes T_{ih}.

4. If $T_{ih} < 0$ the swap is carried out, D_p *and* E_p *are updated* for every object p, and we return at Step 1. If $\min T_{ih} > 0$, the value of the objective cannot be decreased and the algorithm halts. Of course, this happends when all values of T_{ih} are positive and this is precisely the halting condition of the algorithm.

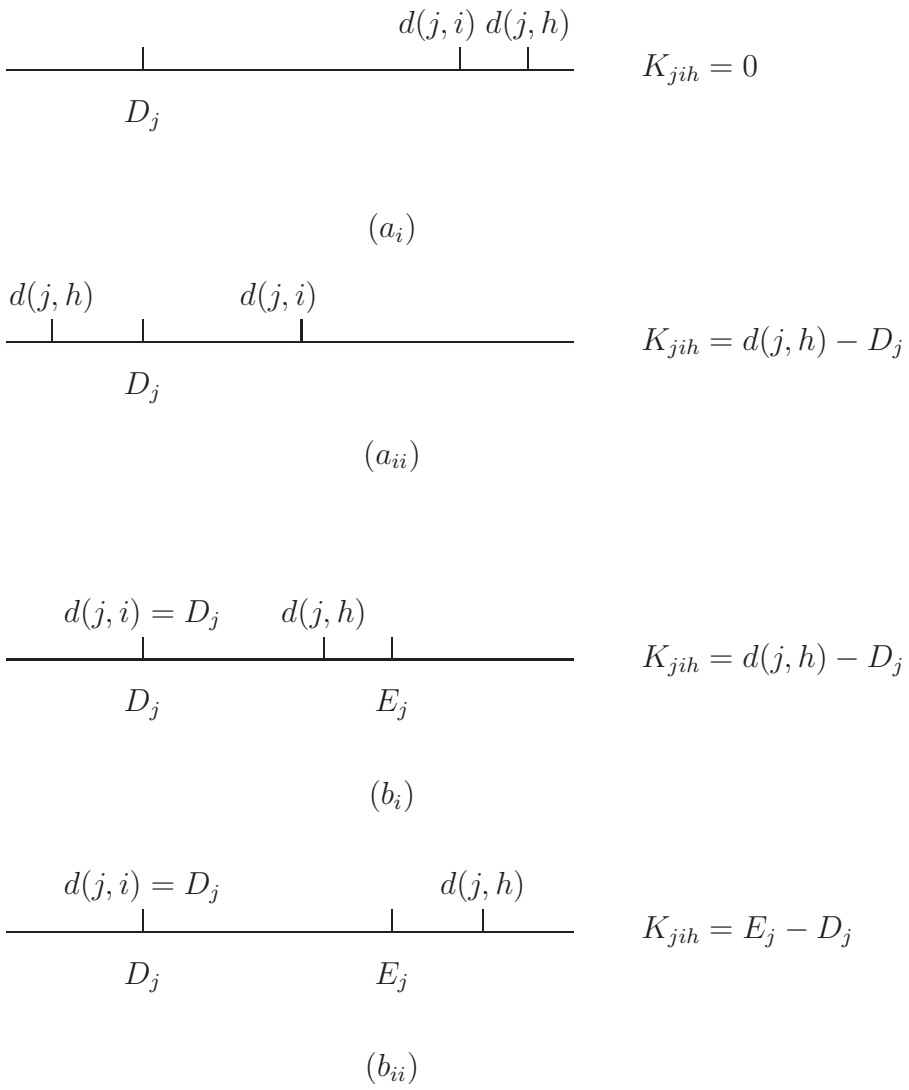

Figure 2: Computation of the contribution K_{jih} of object j to the swap of $i \in S$ with $u \in U$