# A Deterministic k-means Initialization Method

O. Kettani
F. Ramdani

Scientific Institute Mohammed V University in Rabat, Morocco

**Abstract.** The most prominent clustering algorithm k-means has a major drawback: it's sensitivity to the initial clustering centers. To overcome this problem, we propose to initialize k-means by using the Agglomerative Clustering Method (ACM) introduced by the authors in a previous work. The complexity of the proposed approach is O(nk), where n is the number of objects in the input dataset and k the number of clusters. We evaluated its performance by applying on various benchmark datasets and comparing with the related Katsavounidis, Kuo and Zhang (KKZ) O(nk) algorithm. Experimental results have demonstrated that the proposed approach produces more consistent clustering results in term of average Silhouette index.

**Key words:** k-means, clustering, KKZ, initialization.

## Introduction

Cluster analysis is the most widely used technique in Data Mining. Clustering consists of grouping a given dataset into a predefined number of disjoint sets, called clusters, so that the elements in the same cluster are more similar to each other and more different from the elements in the other cluster. This optimization problem is known to be NP-hard, even when the clustering process deals with only two clusters (Aloise et al. 2009: 245-249). Therefore, many heuristics have been proposed, in order to find near optimal clustering solution in reasonable computational time. The most prominent clustering method k-means is a greedy algorithm which has two stages: Initialization, in which we set the seed set of centroids, and an iterative stage, called Lloyd's algorithm (Lloyd, 1982: 129-137). Additionally, Lloyd's algorithm has two steps: The assignment step, in which each object is assigned to its closest centroid, and the centroid's update step. The time required for the assignment step is O(nkd), while the centroid's update step and the computation of the error function is O(nd). The main advantage of k-means is its fast convergence to a local minimum. A major drawback of k-means is its sensitivity to the initial clustering centers (namely, seed). To achieve a better initialization, many techniques have been proposed. In this study, yet another k-means initialization technique is proposed. It is based on the Agglomerative Clustering Method (ACM) introduced by Kettani et al. (2014: 1-7).

## Related Works

Several initialization methods have been proposed in the literatures (MacQueen, 1967: 281-297; Katsavounidis et al., 1994: 144-146). Katsavounidis et al. (1994: 144-146) utilize the sorted pairwise distances for initialization which has been termed as the KKZ algorithm. This algorithm chooses the vector with maximal norm as the first seed, then For j = 2, . . . , k, each centroid $m_j$ is set in the following way: For any remaining data $x_i$ , its distance $d_i$ to the existing centroids is computed. $d_i$ is calculated as the distance between $x_i$ to its closest existing centroid. Then, the point with the largest $d_i$ is selected as $m_j$. The computational complexity of KKZ is O(nk).

## Proposed Approach

The main idea of the proposed ACM method, is to choose k initial centroids as the k first points in X. Then, X is scanned once, the distance between each non assigned point $X_i$ and the nearest centroid $m_j$ is compared with the minimum of the inter-cluster distances stored in a kxk D matrix. If it is lower, then $X_i$ is assigned to cluster $C_j$, else the two clusters with closest centroids are merged together into one cluster and a singleton cluster is created with centroid $X_i$, seeking to minimize the SSE criterion. Then centroids and D matrix are updated. This process is repeated until all points in X are assigned. More details are presented in the following pseudo-code (Table 1).

Table 1. Pseudo-code of the Proposed Method

Input: A data set X whose cardinality is n and an integer k
Output: k centroids $m_i$

```
for i=1:k do
    Ci← {Xi}
    mi← Xi
end for
 D←(d(mi,mj))1 ≤ i≠j ≤ k
mu←Min(D) and (a,b)←Arg(Min(D))
          i,j                    i,j

For i=k+1:n do
        di←Min(d(Xi,mj))
             j
        c←Arg(Min d(Xi,mj))
                      j
           if di<mu then
             Cc← Cc∪{Xi}
               mc←(|Cc|mc+Xi)/(|Cc|+1)
                     D(c,:)←(d(mc,mj)) 1 ≤ j ≤ k
                     D(:,c)←D(c,:)'
         else
              Ca← Ca∪Cb
             ma←(|Ca|ma+|Cb|mb)/(|Ca|+|Cb|)
               Cb←{Xi}
              mb ←Xi
             D(a,:)←(d(ma,mj)) 1 ≤ j ≤ k
                     D(:,a)←D(a,:)'
                      D(b,:)←(d(mb,mj)) 1 ≤ j ≤ k
              D(:,b)←D(b,:)'
          end if
          mu←Min(D) and (a,b)←Arg(Min(D))
           h,j                    h,j
  end For
For i=1:k do
   Output mi
end For
```

*Complexity*

As shown in pseudo-code, at step 3, $O(k^2)$ operations are required to compute D matrix, and $O(k^2)$ space are required to store D matrix.

At step 4, the for loop is repeated n-k times, and updating D matrix, require only O(k) operations at each iteration. The overall running time complexity of ACM is O(nk) which corresponds to the complexity of KKZ method.

**Results**

Algorithm validation is conducted using different data sets from the UCI Machine Learning Repository (Asuncion. and Newman, 2007). We evaluated its performance by applying on several benchmark datasets and compare with KKZ_ k-means. In preprocessing step, the data were normalized.

Silhouette index which measures the cohesion based on the distance between all the points in the same cluster and the separation based on the nearest neighbor distance, was used in these experiments in order to evaluate clustering accuracy.

The silhouette width silh(i ) ranges from -1 to 1. If an observation has a value close to 1, then the data point is closer to its own cluster than a neighboring one. If it has a silhouette width close to -1, then it is not very well clustered. Kaufman and Rousseeuw (2005) use the average silhouette width to estimate the number of clusters in a data set by using the partition with two or more clusters that yields the largest average silhouette width.

ACM_k-means was compared with a related deterministic clustering method: KKZ_k-means (k-means initialized by KKZ).

Experimental results are reported in Table 2 and Fig. 1.

Table 2. Experimental results of KKZ_k-means and ACM_k-means application on different datasets in term of average Silhouette value

| Data set | k | KKZ_k-means | ACM_k-means |
|---|---|---|---|
| Iris | 3 | 0.7525 | 0.8139 |
| Ruspini | 4 | 0.9097 | 0.9107 |
| Aggregation | 7 | 0.6719 | 0.7874 |
| Compound | 6 | 0.6516 | 0.7685 |
| Pathbased | 3 | 0.7330 | 0.7274 |
| Spiral | 3 | 0.5280 | 0.5218 |
| D31 | 31 | 0.6204 | 0.9182 |
| R15 | 15 | 0.5966 | 0.9355 |
| Jain | 2 | 0.6723 | 0.9080 |
| Flame | 2 | 0.5329 | 0.8756 |
| Dim32 | 16 | 0.7470 | 0.9961 |
| Dim64 | 16 | 0.9985 | 0.9984 |
| Dim128 | 16 | 0.9991 | 0.9991 |
| Dim256 | 16 | 0.9996 | 0.9996 |
| Dim512 | 16 | 0.9998 | 0.9998 |
| dim2 | 9 | 0.7816 | 0.9179 |
| dim3 | 9 | 0.3966 | 0.9495 |
| dim4 | 9 | 0.5849 | 0.9749 |
| dim5 | 9 | 0.4490 | 0.9273 |
| dim6 | 9 | 0.6308 | 0.9557 |

| | | | |
|---|---|---|---|
| dim7 | 9 | 0.5652 | 0.9553 |
| dim8 | 9 | 0.4604 | 0.9184 |
| dim9 | 9 | 0.3778 | 0.9941 |
| dim10 | 9 | 0.3729 | 0.9937 |
| dim11 | 9 | 0.5092 | 0.9943 |
| dim12 | 9 | 0.4329 | 0.9924 |
| dim13 | 9 | 0.6241 | 0.9920 |
| dim14 | 9 | 0.6909 | 0.9922 |
| dim15 | 9 | 0.7210 | 0.9087 |
| a1 | 20 | 0.5527 | 0.9905 |
| a2 | 35 | 0.5907 | 0.7421 |
| a3 | 50 | 0.5889 | 0.7703 |
| s1 | 15 | 0.7333 | 0.8805 |
| s2 | 15 | 0.6127 | 0.8009 |
| s3 | 15 | 0.6225 | 0.6481 |
| s4 | 15 | 0.6019 | 0.6075 |

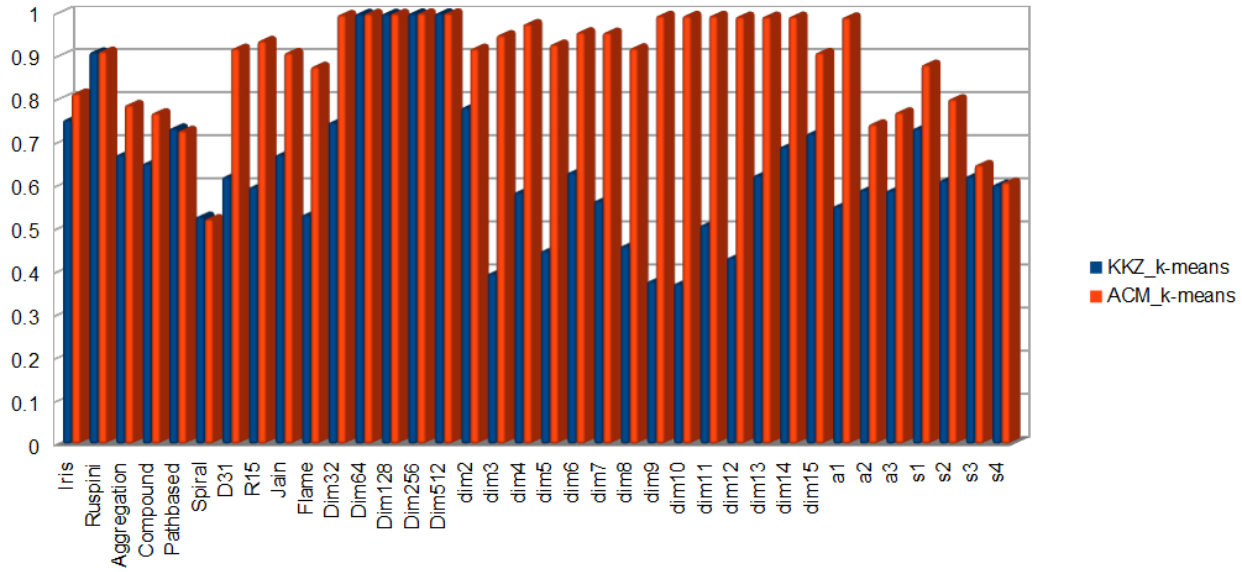


Fig. 1. Chart of average Silhouette index for KKZ_k-means and ACM_k-means applied on different datasets

**Conclusion**

In this study, an initialization method for the k-means algorithm was suggested. Its time complexity is 0(nk) like the KKZ seed algorithm. However, experimental results have demonstrated that the proposed approach produces more consistent clustering results in term of average Silhouette index.

**References**

Aloise, D., Deshpande, A., Hansen, P., Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. Machine Learning, 75, 245-249. https://doi.org/10.1007/s10994-009-5103-0

Asuncion, A., Newman, D.J. (2007). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Available at: http://www.ics.uci.edu/~mlearn/MLRepository.html

Katsavounidis, I., Jay Kuo, C. C., Zhang, Z. (1994). A new initialization technique for generalized lloyd iteration. IEEE Signal Processing Letters, 1, 144-146. https://doi.org/10.1109/97.329844

Kaufman, L., Rousseeuw, P. (2005). Finding groups in data: an introduction to cluster analysis. Wiley.

Kettani, O., Ramdani, F., Tadili, B. (2014). An Agglomerative Clustering Method for Large Data Sets. International Journal of Computer Applications, 92(14), 1-7. https://doi.org/10.1002/9780470316801

Lloyd, S.P. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129-137. https://doi.org/10.1109/TIT.1982.1056489

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, Vol. 1: Statistics, pp. 281-297. Available at: https://projecteuclid.org/euclid.bsmsp/1200512992