On Careful Selection of Initial Centers for K-means Algorithm

R. Jothi, Sraban Kumar Mohanty and Aparajita Ojha

Abstract K-means clustering algorithm is rich in literature and its success stems from simplicity and computational efficiency. The key limitation of K-means is that its convergence depends on the initial partition. Improper selection of initial centroids may lead to poor results. This paper proposes a method known as Deterministic Initialization using Constrained Recursive Bi-partitioning (DICRB) for the careful selection of initial centers. First, a set of probable centers are identified using recursive binary partitioning. Then, the initial centers for K-means algorithm are determined by applying a graph clustering on the probable centers. Experimental results demonstrate the efficacy and deterministic nature of the proposed method.

Keywords Clustering · K-means algorithm · Initialization · Bi-partitioning

1 Introduction

Clustering is the process of discovering natural grouping of objects so that objects within the same cluster are similar and objects from different clusters are dissimilar according to certain similarity measure. Various methods for clustering are broadly

R. Jothi (✉) · S.K. Mohanty · A. Ojha
Indian Institute of Information Technology, Design and Manufacturing Jabalpur,
Jabalpur, Madhya Pradesh, India
e-mail: r.jothi@iiitdmj.ac.in
URL: http://www.iiitdmj.ac.in

S.K. Mohanty
e-mail: sraban@iiitdmj.ac.in
URL: http://www.iiitdmj.ac.in

A. Ojha
e-mail: aojha@iiitdmj.ac.in
URL: http://www.iiitdmj.ac.in

A. Nagar et al. (eds.), *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, Smart Innovation, Systems and Technologies 43, DOI 10.1007/978-81-322-2538-6_45

classified into hierarchical and partitional methods [1, 2]. Hierarchical methods generate a nested grouping of objects in the form of dendrogram tree. Single-linkage and complete-linkage are the well known hierarchical clustering methods. The major drawback of these algorithms is their quadratic run time which poses a major problem for large datasets [3]. In contrast to hierarchical methods, partitional methods directly divide the set of objects into k groups without imposing the hierarchical structure [1].

K-means is a popular partitional clustering algorithm with the objective of minimizing the sum of squared error (SSE) which is defined as the sum of the squared distance between the cluster centers and the points in the cluster. K-means starts with k randomly chosen initial centers and assign each object in the dataset to a nearest center. Iteratively, the centers are recomputed and objects are reassigned to the nearest centers. Due to its simple implementation, K-means has been extensively used in various scientific applications. However, the results of K-means strongly depend on the choice of initial centers [1, 3, 4].

If the initial centers are improperly chosen, then the algorithm may converge to a local optimum. Number of approaches have been proposed to wisely choose the initial seeds for K-means [5–10].

A sampling based clustering solution for initial seed selection was suggested by Bradley and Fayyad [5]. The idea was to choose several samples from the given set of objects and applying K-means on each of the sample independently with random centers. The resulting centroids from each subcluster is the potential guess of the centers for the whole dataset. Likas et al. [8] proposed global K-means algorithm which incrementally chooses k cluster centers one at a time. Experimental results demonstrated that their method outperforms the K-means algorithm. But, this method is computationally expensive as it requires N execution of the K-means algorithm on the entire dataset, where N is the number of objects in the dataset [4].

Arthur and Vassilvitskii proposed an improved version of K-means known as K-means++ [7]. It chooses the first center c_1 randomly from the dataset and other centers c_i, $2 \leq i \leq k$ are chosen such that distance between c_i and the previously chosen centers is maximum. Both theoretically and experimentally they have shown that, K-means++ not only speed up the convergence of the clustering process but also yields a better clustering result than K-means.

Alternatively, may initialization methods were discussed by considering the principal dimensions for splitting the dataset [6, 9]. Ting and Jennifer [9] proposed two divisive hierarchical approaches, namely PCA-part method and Var-part method which identify the centers for K-means algorithm using k-splits along the better discriminant hyperplanes.

Erisoglu et al. [6] proposed a method which first defines the subspace of the dataset X along two main dimensions that best represent the spread of the dataset. Then, the data point with the longest distance from the centroid of X in the subspace is chosen as the first cluster center. The subsequent centers are chosen such that the center c_i has the maximum distance from the previously computed centers $c_1, \ldots, c_{i-1}$.

Min-Max algorithm proposed by Tzortzis and Likas [10] tackles initialization problem by implementing a weighted version of the K-means by assigning weights to the clusters in proportion to their variance. This weighting scheme attains high quality partition by controlling the clusters with larger variance.

There are various methods for cluster initialization, a comparative study of which can be seen in [4]. However, many of the methods run in quadratic time [8, 11] which degrades the efficiency of K-means. In this paper, we propose an initialization method which runs in $O(n \lg n)$ time using constrained recursive bi-partitioning. The performance of the proposed algorithm has been demonstrated using experimental analysis.

The rest of the paper is organized as follows. The description of K-means algorithm is given in Sect. 2. The proposed method is explained in Sect. 3. The complexity of the proposed method is discussed in Sect. 4. The experimental analysis is shown in Sect. 5. The conclusion and future scope are given in Sect. 6.

2 K-Means Algorithm

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of objects to be clustered into k groups, where k is the number of classes of objects, which must be known a priori. The objective of K-means is to find a k-partition of X such that the sum of squared error (SSE) criterion is minimized. Let $S = \{S_1, S_2, \ldots S_k\}$ be the set of partitions returned by K-means and let μ_i be the center of the partition s_i. The sum of squared error (SSE) is defined as follows [1].

$$SSE = \sum_{i=1}^{k} \sum_{x_j \epsilon s_i} d(x_j, \mu_i)^2. \tag{1}$$

where $d(\ldots)$ denotes the distance (dissimilarity). K-means algorithm starts with k arbitrary centers and iteratively recomputes the centers and reassigns the points to the nearest centers. If there is no change in centers, then the algorithm stops. The K-means algorithm is described as follows.

Algorithm 1 *K-means Algorithm [2]*

```
Input: Dataset $X$.
Output: $k$ clusters of $X$.

1. Randomly choose $k$ centers $\mu_i$, $1 \leq i \leq k$.
2. For each object $x_j$ in $X$, compute distance between $x_j$ and $\mu_i$, $1 \leq i \leq k$.
3. Assign $x_j$ to the partition $s_i$, such that the distance between $x_j$ and $\mu_i$ is minimum.
4. Recompute centers; repeat steps 2 and 3 if there is change in centers. Else stop.
```

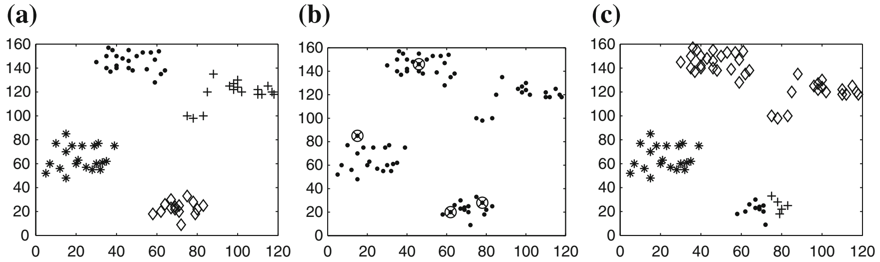

Fig. 1 K-means converging to a local optimum with random center initialization. **a** A given dataset with four clusters. **b** Randomly chosen initial centers. **c** The result of K-means with centers chosen in (**b**)

As the initial centers for K-means partition are chosen randomly, two or more centers may collide in a nearby region. As a consequence, the points are forced to be assigned to one of the nearest centers, leading to poor results. This is illustrated in Fig. 1. It is clear from the figure that, K-means gets trapped with local optimum if the initial centers are not chosen properly.

3 Proposed Method

The proposed method is a two-step process. In the first step, it identifies a set of probable centers by dividing the dataset X into $k^{'}$ partitions. During the second step, the actual centers are determined by grouping the probable centers into k subsets, where center of each subset is considered as one of the k centers for K-means algorithm.

3.1 Identifying Probable Centers

K-means algorithm tries to assign the points to a cluster based on a nearest center. Representing a cluster using a single prototype (center) may not capture the intrinsic nature of clusters [12]. Motivated from [12], we identify a pool of $k^{'}, k^{'} \gg k$ most likely points from the dataset, such that the spread of each cluster in the dataset is well represented by a subset of these points. We call these points as *probable centers* denoted by $Y = \{y_1, y_2, \ldots, y_{k'}\}$. In order to identify a set of probable centers, this paper proposes an algorithm known as Constrained Bi-partitioning algorithm.

Constrained Bi-partitioning Algorithm Generally a Bi-partitioning method recursively split the dataset into a binary tree of partitions [13]. It starts from the root node that contains the given dataset X and split the node into two subsets X_1

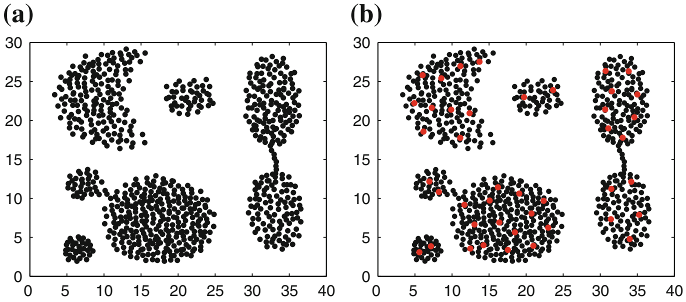

Fig. 2 The probable centers. **a** The dataset. **b** The probable centers are marked in *red dots* (color figure online)

and X_2 based on certain partitioning criteria. The recursive splitting stops once k partitions are identified.

This paper proposes an improved version of Binary partitioning known as Constrained Bi-partitioning algorithm. The basis of bipartitional criteria adapted in our proposed algorithm lies in choosing the two centers for splitting a node in the tree. Two centers p and q are chosen such that they are the farthest pair of points in the node in order to maximize the inter-cluster variance. While Bi-partitioning algorithm stops when k partitions are identified, the constrained algorithm continues the splitting process as long as the size of the subset to be partitioned is greater than $\sqrt{n}$. Thus, the number of partitions is not preset. The leaf nodes, the subsets which cannot be partitioned further, are stored in the set of partitions $S = \{S_1, S_2, \ldots, S_{k'}\}$. The center of each partition S_i in S is considered to be a probable center y_i.

Figure 2 shows an example of a dataset and its probable centers. It is obvious from the figure that, each cluster is covered by a subset of probable centers. The actual centers of the K-means algorithm can be identified by merging these probable centers into k groups.

3.2 Computation of k Initial Centers from k' Probable Centers

Once the set of probable centers $Y = \{y_1, y_2, \ldots, y_{k'}\}$ are recognized, next we need to identify the k disjoint subsets of Y. The probable centers must be grouped such that the closeness between the centers within a subset is high as compared to the closeness between the centers of different subsets. This in turn maximizes the inter-cluster separation of the final clusters produced by the K-means algorithm.

The relative inter-connectivity of the probable centers intuitively expresses the neighboring nature of the subsets and the breaks in the connectivity gives a clue on the separation of actual clusters in the dataset. In order to identify such breaks, we

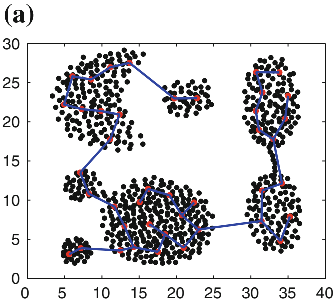

Fig. 3 MST-based partitioning of probable centers for the dataset shown in Fig. 2a. **a** The MST of probable centers. **b** k disjoint subsets of probable centers

employ Minimum Spanning Tree (MST)-based representation of probable centers, as MST of a set of points can be used to reflect the similarity of the points with their neighborhood [14]. Simply removing $k-1$ longest edges from the MST results in k disjoint subsets of nearest centers, such that each subset would represent a cluster. This is illustrated in Fig. 3.

As each $y_i \epsilon Y$ may or may not belong to the dataset X, we choose a best representative point $r_i \epsilon X$ from each S_i such that r_i is closest to y_i. Let $R = \{r_1, r_2, \ldots, r_{k'}\}$ denotes the set of best representative points identified in the above manner. Prim's algorithm on R generates MST T_1, removing $k-1$ longest edges from T_1 yields k clusters of best representative points. Finally, the actual centers for K-means algorithm are computed from the center of the k clusters. The proposed method DICRB is summarized in Algorithm 2. The result of K-means initialized with the proposed method is demonstrated in Fig. 4.

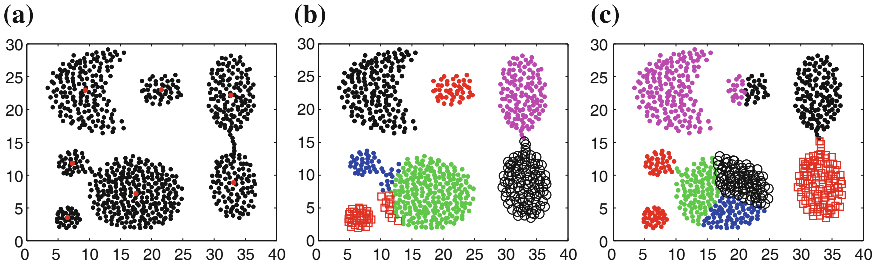

Fig. 4 Result of K-means initialized with proposed method. **a** The centers identified by proposed method marked in *red dots*. **b** The final clusters identified by K-means initialized with proposed method. **c** Result of K-means with random initialization (color figure online)

Algorithm 2 *Deterministic Initialization using Constrained Recursive Bi-partitioning (DICRB) Algorithm*

```
Input: Dataset $X$.
Output: $k$ initial centers for K-means algorithm.

1. Let $S$ be the set of partitions and initialize $S = \phi$.
2. Initialize the node to be splitted $X' = X$.
3. Repeat
   3.1 if size of $X' > \sqrt{n}$
      3.1.1 Find the center $\mu$ of the node $X'$.
      3.1.2 Find a point $o \epsilon X'$ such that $d(o, \mu)$ is minimum.
      3.1.3 Choose two centers $p \epsilon X'$ and $q \epsilon X'$ such that $p$ is farthest point from $o$ and
            $q$ is farthest point from $p$.
      3.1.4 Split the node $X'$ into two nodes $X_p$ and $X_q$ according to centers $p$ and $q$.
      3.1.5 Recursively apply bi-partitioning on $X_p$ and $X_q$.
   3.2 else $S = \{S \cup X'\}$
4. Until there is no node to split
5. Identify a set of probable centers $Y = \{y_1, y_2, \cdots, y_{k'}\}$, where $y_i$ is the center of
   the subset $s_i$.
6. Build a set of best representative points $(R)$ by choosing points closer to each prob-
   able center $y_i \epsilon Y$.
7. Construct MST $T_1$ of $R$.
8. Remove $k-1$ longest edges from $T_1$ to get $k$ clusters.
9. Center from each of the $k$ clusters corresponds to actual center for K-means algo-
   rithm.
```

4 Complexity of the Proposed Method

The complexity of the DICRB method is analyzed as follows. The steps 1–4 take $O(n \lg n)$ time to construct binary partitioning tree. Step-5 takes $O(n)$ time to identify the probable centers from each partition. Similarly $O(n)$ time is needed to find the best representative point set R in step-6. As the size of the set R is $O(\sqrt{n})$, Prim's algorithm in step-7 takes $O(n)$ and clustering in step-8 takes $O(\sqrt{n})$ complexity. Hence, the overall time complexity to identify the initial cluster centers for K-means algorithm is $O(n \lg n)$.

5 Experimental Results

The proposed method of initialization is compared against random initialization based on the number of iterations (I) required for convergence, *SSE* and Adjusted Rand Index (ARand) [15] of clusters after the convergence. The tests are conducted on four synthetic and four real datasets. The K-means algorithm with randomly chosen initial centers and K-means algorithm with proposed method of initialization are run for 100 times on the identical datasets and the average value of *I*, *SSE* and ARand are observed. We also report the maximum and minimum number of iterations taken by both the methods.

The synthetic datasets DS1, DS2, DS3 and DS4 used in our experiments are chosen such that each dataset would represent different kind of clustering problem. The details of these datasets are given in Table 1 and are shown in Fig. 5. The results of K-means on these datasets are shown in Tables 2 and 3. It is evident from the results provided in the Tables 2 and 3 that, K-means initialized with proposed method performs better in terms of constant number iterations and improved cluster quality with respect to internal as well as external quality measures.

We have observed the results of proposed method also on few real datasets taken from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). The details of these datasets can be seen in Table 1. Our proposed initialization method maintains the stable results from K-means according to the number of iterations, minimized error and improved cluster separation as compared to K-means with random centers. This is evident from the results shown in Tables 4 and 5.

Table 1 Details of the datasets. No. of instances (*n*), no. of dimensions (*d*), no. of clusters (*k*)

2-*Dimensional synthetic datasets*			
Dataset	*n*		*k*
DS1	399		6
DS2	788		7
DS3	600		15
DS4	300		2
Real datasets			
Dataset	*n*	*d*	*k*
Iris	150	5	3
Ruspini	75	2	4
WDBC	569	32	2
Thyroid	215	5	6

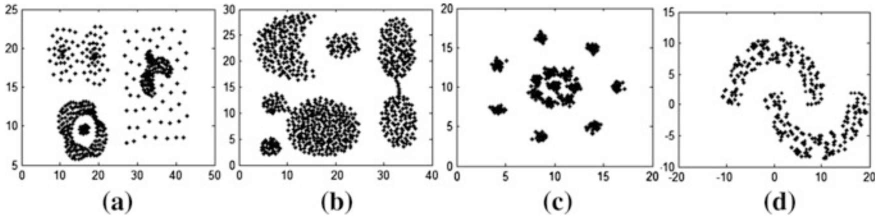

Fig. 5 Synthetic datasets used for experimental study

Table 2 Comparison of initialization methods according to number of iterations (I) on synthetic datasets

Dataset	Method	Avg (I)	Max (I)	Min (I)
DS1	Random	13	39	5
	Proposed	**9**	**9**	**9**
DS2	Random	16	33	6
	Proposed	**10**	**10**	**10**
DS3	Random	1	20	4
	Proposed	**6**	**6**	**6**
DS4	Random	8	13	4
	Proposed	**6**	**6**	**6**

Table 3 Comparison of initialization methods according to SSE and adjusted rand index of clusters on synthetic datasets

Dataset	Method	SSE	Adjusted rand
DS1	Random	4801.04	0.6936
	Proposed	**4733.10**	**0.8361**
DS2	Random	12322.38	0.7967
	Proposed	**11514.35**	**0.9277**
DS3	Random	1001.47	0.8485
	Proposed	**186.57**	**0.9091**
DS4	Random	11885.37	0.6229
	Proposed	**11876.01**	**0.6536**

Table 4 Comparison of initialization methods according to number of iterations (I) on real datasets

Dataset	Method	Avg (I)	Max (I)	Min (I)
Iris	Random	9	15	3
	Proposed	**6**	**6**	**6**
Ruspini	Random	4	9	2
	Proposed	**2**	**2**	**2**
WDBC	Random	10	14	8
	Proposed	**8**	**8**	**8**
Thyroid	Random	9	15	3
	Proposed	**3**	**3**	**3**

Table 5 Comparison of initialization methods according to SSE and adjusted rand index of clusters on real datasets

Dataset	Method	SSE	Adjusted rand
Iris	Random	106.35	0.8749
	Proposed	**87.31**	**0.9799**
Ruspini	Random	29385.95	0.7360
	Proposed	**13269.65**	**1.000**
WDBC	Random	11689.49	0.6972
	Proposed	**11640.71**	**0.7988**
Thyroid	Random	528.28	0.5608
	Proposed	**502.11**	**0.5907**

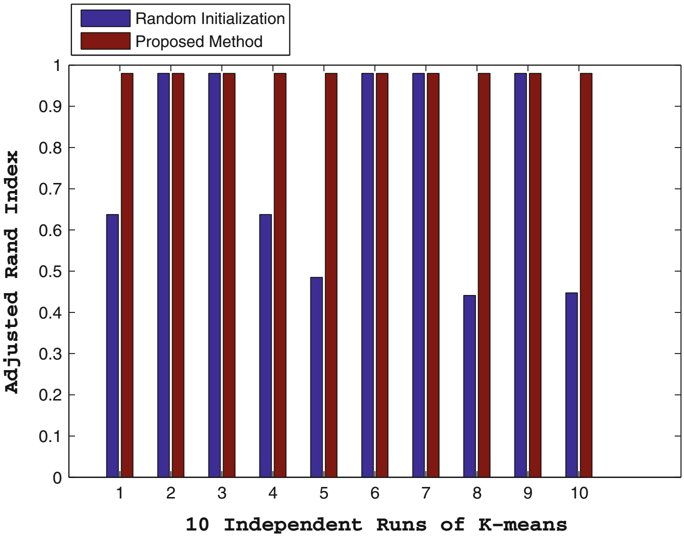

Fig. 6 Adjusted rand index of random and proposed method on iris dataset in each run

With random initialization method, the initial centers are arbitrary in each run and thus the final clustering results are not deterministic. With the proposed method, the initial centers are chosen from a pool of probable candidate centers which always produce deterministic clustering results. The Fig. 6 shows the Adjusted Rand value obtained from random initialization and proposed method on Iris dataset for 10 runs of K-means. As can be seen from the figure, the Adjusted Rand value for random initialization method changes with every run where as it remains constant in the proposed method, showing the deterministic nature of our algorithm.

6 Conclusion

This paper proposed an initialization method for K-means algorithm using constrained recursive bi-partitioning. The efficiency of the proposed method was demonstrated through experiments on different synthetic and real datasets. While the clustering results of K-means algorithm with random initialization is unstable, our proposed method of initialization produces deterministic results. As a future work, we will carry out an extensive analysis of DICRB method on more datasets.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
2. Han, J., Kamber, M.: Data Mining, Southeast Asia Edition: Concepts and Techniques. Morgan Kaufmann, Los Altos (2006)
3. Xu, R., Wunsch, D., et al.: Survey of clustering algorithms. IEEE Trans. Neural Networks **16** (3), 645–678 (2005)
4. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013)
5. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. ICML **98**, 91–99 (1998)
6. Erisoglu, M., Calis, N., Sakallioglu, S.: A new algorithm for initial cluster centers in k-means algorithm. Pattern Recogn. Lett. **32**(14), 1701–1705 (2011)
7. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
8. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern Recogn. **36**(2), 451–461 (2003)
9. Ting, S., Jennifer, D.G.: In search of deterministic methods for initializing k-means and gaussian mixture clustering. Intell. Data Anal. **11**(4), 319–338 (2007)
10. Tzortzis, G., Likas, A.: The minmax k-means clustering algorithm. Pattern Recogn. **47**(7), 2505–2516 (2014)
11. Cao, F., Liang, J., Jiang, G.: An initialization method for the k-means algorithm using neighborhood model. Comput. Math. Appl. **58**(3), 474–483 (2009)
12. Liu, M., Jiang, X., Kot, A.C.: A multi-prototype clustering algorithm. Pattern Recogn. **42**(5), 689–698 (2009)
13. Chavent, M., Lechevallier, Y., Briant, O.: DIVCLUS-T: a monothetic divisive hierarchical clustering method. Comput. Stat. Data Anal. **52**(2), 687–701 (2007)
14. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. **100**(1), 68–86 (1971)
15. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. J. Intell. Inf. Syst. **17**(2), 107–145 (2001)