Centroids Initialization for K-Means Clustering using Improved Pillar Algorithm

Bapusaheb B. Bhusare[1], S. M. Bansode[2]

ME Student, CSE, Government College of Engineering, Aurangabad (MH), India[1].

Asst. Professor, CSE, Government College of Engineering, Aurangabad (MH), India[2].

Abstract: The K means clustering algorithm generates the initial centroids randomly which does not consider the placement of them spreading in the feature space. The performance of K means highly depends upon the correctness of the initial centroids which are chosen randomly that can be trapped in local minima and led to incorrect clustering results. In this paper we propose a new approach to optimize the initial centroids for K means which spreads them in the feature space uniformly so that the distance among them is as far as possible. Firstly grand mean of all data point is calculated and the accumulated distance metric between each data point and the grand mean is calculated. The data point which has maximum distance will be considered as first initial centroid. This approach also has a mechanism to avoid outlier data being chosen as initial centroids. We also reduce the complexity of our algorithm by excluding the designated initial centroids neighbors from next iterations. The experimental results show the improvement in the solution using the proposed system.

Key words: Clustering, K-means algorithm, Initial centroids, Pillar algorithm, Improved Pillar algorithm.

I. INTRODUCTION

Clustering is widely used knowledge discovery technique to classify unsupervised objects in the same groups by considering their similarities. A good cluster is constructed when the members of the clusters have high degree of similarity of each other (internal homogeneity) and are not like members of other clusters (external homogeneity). It means the process to define a mapping f: D→C from some data D= {d_1, d_2,.., d_n} to some clusters C={c_1,c_2,....,c_n} on similarity between d_i. The application of clustering are in many fields such as data mining, pattern recognition, image classification, biological sciences, document retrievals, etc.

K-means algorithm is popular and widely used partitioning clustering technique which was developed by Mac Queen in 1967. If the randomly selected initial centroids are close to a final cluster center, then K-means clustering can find the accurate clusters. It is not always the case. If selected initial centroids are far from the cluster center, it will lead to incorrect clustering results [1]. Because of initial centroids generated randomly, K-means clustering does not guarantee the unique clustering results [2]. So for a K-means it is difficult to reach global optimum, but only to one of local minima. The better results of K-means clustering can be achieved after executing more than one times. However, it is difficult to decide the execution limit, which gives the best performance [3].

Several methods proposed to solve the cluster initialization for K-means clustering. Duda and Hart discussed a recursive method for initializing the means by running k problems. A variation of this method consists of taking the entire data into account and then randomly perturbing it k times [2] Shehroz and Ahmad, (2004). Bradley and Fayyad (1998) proposed an algorithm that refines initial points by analyzing distribution of the data and probability of data density [4]. Pena et al. (1999) presented empirical comparisons for four initialization methods for K-means clustering those are random, Forgy approach, Mac Queen Approach and Kaufman approach [5]. Barakbah and Helen (2005) presented an algorithm, called as Optimized K-means, that spreads the initial centroids in the feature space so that distance among them are as far as possible [6].Ali Ridho Barakbah (2006) proposed a new algorithm to optimize the initial centroids for K-means by separately locating them as far as possible in data distribution [7]. Arai and Barakbah (2007)

proposed a Hierarchical K-means algorithm for centroids initialization for K-means [3]. Barakbah and Yasushi Kiyoki (2009) proposed K-means optimization by distance maximization for initial centroid designation [8].

In this paper we propose a new approach for optimizing the initial centroids for K-means, inspired by designation of pillars placement in a house or building. We consider the pillars which should be located as far as possible from each other to withstand against the pressure distribution of a roof, as the number of centroids amongst the data distribution. Therefore, our proposed approach determines the position of initial centroids by calculating the accumulated distance metric between each data and all previous centroids, and then, a data point which has the maximum distance will be selected. This approach is able to locate all centroids separately as far as possible between the initial centroids in the data distribution. This algorithm also has a mechanism to avoid outlier data being chosen as the initial centroids.

II. BASIC THEORY OF K-MEANS

Let A = $\{a_i \mid i=1 \ldots n\}$ be attributes of n-dimensional vector and X = $\{x_i \mid i=1,\ldots,r\}$ be each data of A. The K-means clustering separates X into K partitions called clusters S=$\{s_i \mid i=1,\ldots,k\}$ where M ∈ X is the M=$\{m_i \mid i=1,\ldots,n(s_i)\}$ as member of S. Each cluster has center of C = $\{c_i \mid i=1 \ldots k\}$.

K-means clustering algorithm is described as follows:

1. Initiate its algorithm by generating random starting points of initial cluster centers c_k.
2. Calculate the distance *d(x, c)* between vector x_i to cluster center c_k. Euclidean distance is commonly used to express distance.
3. Separate x_i into S which has minimum *d(x, C)*.
4. Determine the new cluster centers c_i for *i=1,…,k* defined as :

   $c_i = \frac{1}{n} \sum_{j=1}^{n(si)} m_{ij} \in s_i$
5. Go back to step 2 until all centroids are convergent.

   The centroids can be said converged if their positions do not change in the iteration. It may also stop in the *t* iteration with a threshold ε if those positions have been updated by the distance below ε

   $|\frac{c^t - c^{t-1}}{c^t}| < \varepsilon$

III. PILLAR ALGORITHM

A. Basic Concept

The pillar algorithm proposed a method of placing the initial centroids whereby each of them has a farthest accumulated distance between them. So we consider the pillars which should be located as far as possible from each other to with stand against the pressure distribution of a roof, as number of centroids among the gravity weight of data distribution in the vector space. Therefore, our proposed approach in this paper designates positions of initial centroids in the farthest accumulated distance between them in the data distribution.

B. Determining Initial Centroids

First of all, the grand mean of data points is calculated as the gravity center of the data distribution. The distance metric D (let D^1 be D in this early step), is then created between each data point and the grand mean. A data point with highest distance in D^1 will be selected as the first candidate of the initial centroid **x**. Fig. 3a illustrates m as the grand mean of data point and **x** which has the farthest distance to m is the candidate of the first initial centroid. If **x** is not an outlier, it will be promoted to the first initial centroid c_1.

We then calculate D (D^2 in this step), which is the distance metric between each data points and c_1. Starting from this step, we use the accumulated distance metric DM and assign D^2 to DM. the construction of DM is started from D^1. To select a candidate for the second initial centroid, the same mechanism is applied using DM instead of D. The data point with the highest distance of DM will be selected as the second initial centroid candidate **x**, as shown in Fig. 3b. If **x** is not classified as an outlier, it becomes c_2.

To select a next **x** for the candidate of the rest initial centroids, D^i (where i is the current iteration step) is recalculated between each data points and c_{i-1}. The D^i is then added to the accumulated distance metric DM (DM← DM + D^i). This accumulation scheme can avoid the nearest data points to c_{i-1} being chosen as the candidate of the next initial centroid. It consequently can spread out the next initial centroids far away from the previous ones. The iterative process guarantees that all initial centroids are designated.

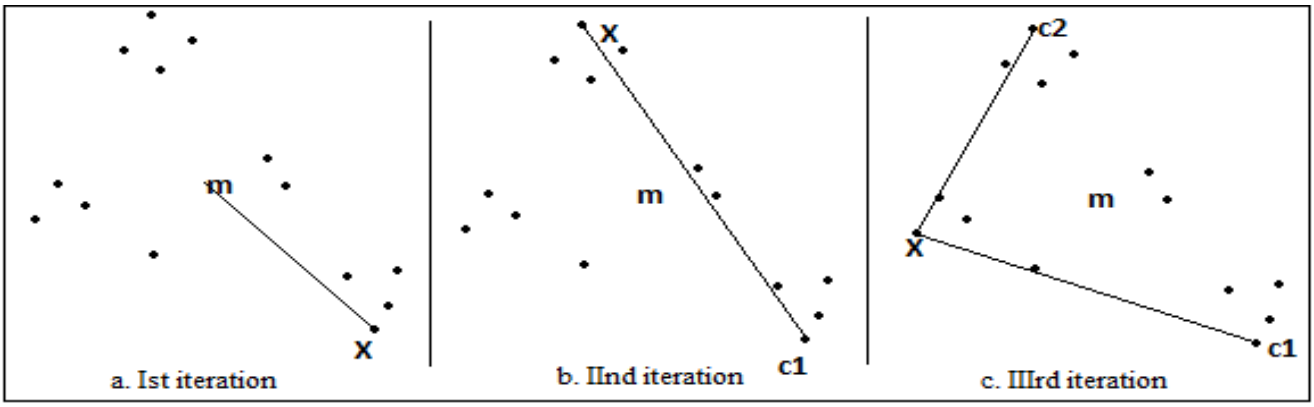


Fig. 3. Selection of several candidates as a initial centroids

C. Outlier Detection Mechanism

To be selected as the initial centroids, the data point candidate must not be categorized as outlier. We identify an outlier by considering the number of neighbor points within the neighborhood boundary. Let n be the number of data points and k be the number of clusters, we set the number of neighbor nmin by using a probabilistic parameter α to the average members of clusters n/k. for assuming the neighborhood boundary nbdis, we apply a threshold β to the highest distance in D_i, as shown in Fig. 4.

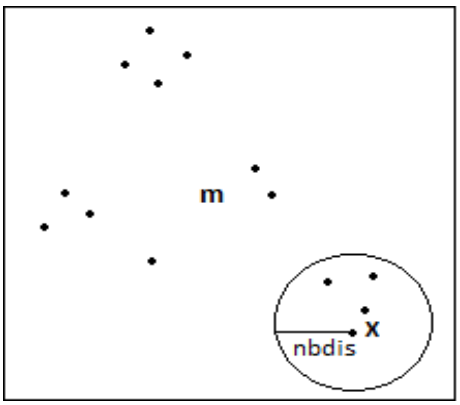


Fig. 4. An illustration of neighborhood boundary

To get several data points to be the neighbors inside the boundary, it needs to calculate a distance metric between all data points and **x**. However, it can utilize distance metric D for the next iteration step. In case shown in Fig. 4, the distance metric between all data points and **x** can be acquired from D^2.

If the number of neighbors inside the boundary is lower than nmin, **x** will be classified as an outlier. When **x** is considered as an outlier, it will be redetermined by selecting the second highest distance in D^1. This outlier detection mechanism is iteratively executed until **x** can be considered as the initial centroid.

D. Pillar Algorithm

Let $X = \{x_i \mid i=1,\ldots,n\}$ be data, k be number of clusters, $C=\{c_i \mid i=1,\ldots,k\}$ be initial centroids, $SX \subseteq X$ be identification for X which are already selected in the sequence of process, $DM = \{x_i \mid i=1,\ldots,n\}$ be accumulated distance metric, $D = \{x_i \mid i=1,\ldots,n\}$ be distance metric for each iteration and m be grand mean of X. The proposed algorithm is described as:

1. Set C = Ø, SX = Ø, and DM=[ ]
2. Calculate D ← dis(X, m)
3. Set number of neighbors nmin = α .n / k
4. Assign dmax ← argmax(D)
5. Set neighborhood boundary nbdis =β. Dmax
6. Set i =1 as counter to determine the i^{th} initial centroid
7. DM = DM + D
8. Select **x** ← $x_{argmax(DM)}$ as the candidate for i^{th} initial centroids
9. SX = SX ∪ **x**
10. Set D as the distance metric between X to **x**.
11. Set no ← number of data points fulfilling D ≤ nbdis
12. Assign DM(**x**) = 0
13. If no < nmin, go to step 8
14. Assign D(SX) = 0
15. C = C ∪ **x**
16. i = i+1
17. Remove all neighbors of selected centroids
18. If i < k, go back to step7
19. Finish in which C is the solution as optimized initial centroids.

IV. IMPROVEMENT IN THE PILLAR ALGORITHM

The pillar algorithm is very effective to position the initial centroids for K-means and improve the precision of the clustering results. However, the algorithm takes highly computation time for clustering huge data which often have many outliers, since its complexity is O((k+h+1) n) where k= number of clusters, h= number of outliers, and n= number of data items to position the initial centroids. We can reduce the complexity of our pillar algorithm by excluding the designated initial centroids neighbors from next iteration so that the complexity will decrease in line with iterations.

A. Excluding initial centroids neighbors.

In the pillar algorithm, when a data item is chosen as initial centroid candidate **x**, the outlier detection mechanism is applied by identifying the neighbors. If the number of neighbors inside the boundary in nbdis, as shown in Fig.4 is same or higher than nmin, **x** will be promoted to an initial centroid c_i in iteration i. In order to designate next initial centroids, we improve the algorithm by reducing number of distance calculation in D^{i+1}. When **x** is promoted to be c_i the neighbors of c_i inside nbdis are noted, these neighbors are supposed to belong to c_i and do not need to involve in the distance calculation D^{i+1}. By excluding them in D^{i+1} number of distance calculations can be reduced for the next steps. It will decrease the complexity and speed up the execution time for designating all initial centroids.

B. Algorithm Complexity

The time complexity of Pillar algorithm is O ((k+h+1) n), where k is number of clusters, n is the number of items and h is number of outliers in the data set. In case if there is no outlier in the data set, the complexity becomes O ((k+1) n), or equal to O ((i+1) n), where i is the number of iterations, since selecting an initial centroid for each k takes one iteration. For worse case in which there are number of outliers close to n (h≈ n), the complexity becomes ≈O (n^2).

With the improvement of Pillar algorithm by excluding the initial centroids neighbors for each designated initial centroids, the number of data n items will decrease in line with the iterations. When the initial centroids neighbors are excluded and not involved in the distance calculation for next steps, n in the iterations will be decreased. The complexity of improved Pillar algorithm is O(n+(h_1. n_1)+…+ (h_k . n_k)) where n_k<…<n_1< n, n_i is the rest of number of data items after excluding the i^{th} designated initial centroids neighbors and h_i is the number of outliers before the i^{th} designated initial centroids. By this improvement, in a case in which there is no outliers, the complexity becomes O ($n+n_1+\ldots+n_k$). The complexity will be ≈O (n^2) for worse case in which number of outliers close to n.

V. EXPERIMENTAL RESULT

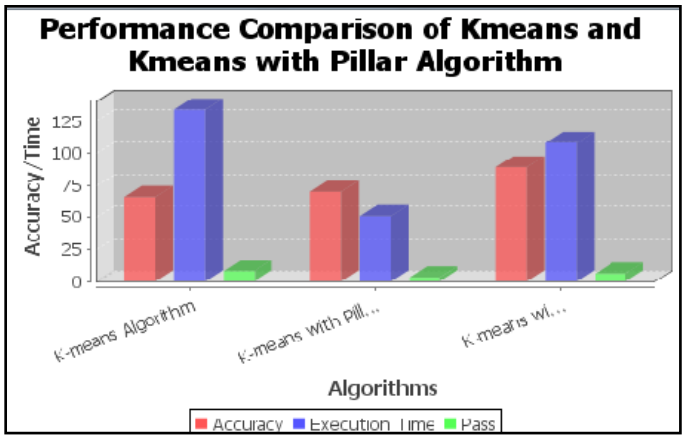
We conducted a series of experiment on different datasets which is obtained from UCI Repository [9]. The datasets are Iris dataset, New Thyroid dataset and Wine dataset. For comparison purpose, we used the plain data of the data set without normalization.

We evaluated the performance comparison between basic K means with random initialization, Pillar K means that optimize the initial centroids for K means by choosing the initial centroids, so that the distance among them is as far as possible and the Improved Pillar K Means which reduce the complexity of our algorithm by excluding the designated initial centroids neighbors from next iterations so that time complexity will decrease in line with iterations and speed up the execution time.

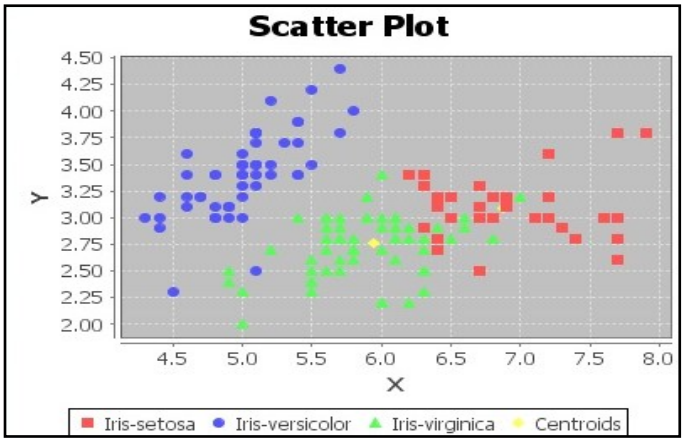
The performance of our proposed algorithm is examined in the accuracy rate, computational time and number of iteration required with different real world dataset and compared with the original K-means algorithm. K means with random centroids initialization cannot give the unique clustering results so we made 10 times experiments and consider the average results. We set $\alpha = 0.25$ and $\beta = 0.60$ for our proposed algorithm to detect outliers. Table I shows the comparison results of our proposed approach with the K means using random initialization. The improved pillar algorithm outperformed the other comparing algorithms. Figure 5 shows the comparison of proposed approach and K Means with accuracy, execution time and number of iterations required for different datasets. Figure 6 shows the clusters scatter plot for different datasets.

Table I Experimental Results

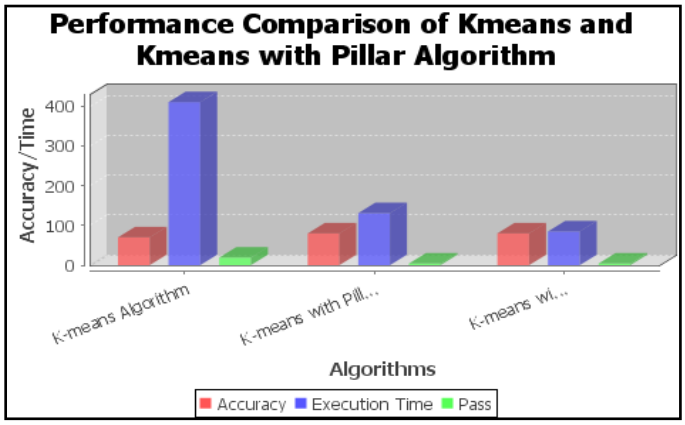
Data set	Iris			Thyroid			Wine		
Algorithms	**K-Means**	**Pillar K-Means**	**Improved Pillar K- Means**	**K-Means**	**Pillar K-Means**	**Improved Pillar K- Means**	**K-Means**	**Pillar K-Means**	**Improved Pillar K- Means**
Accuracy %	66	70	89	70	80	80	69	65	67
Exe.Time (ms)	134	51	108	411	131	85	309	210	244
Iterations	8	3	6	19	4	4	10	5	8



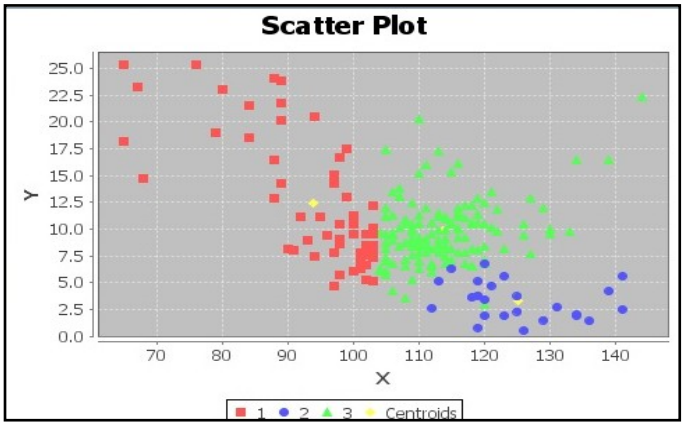
a. Iris dataset



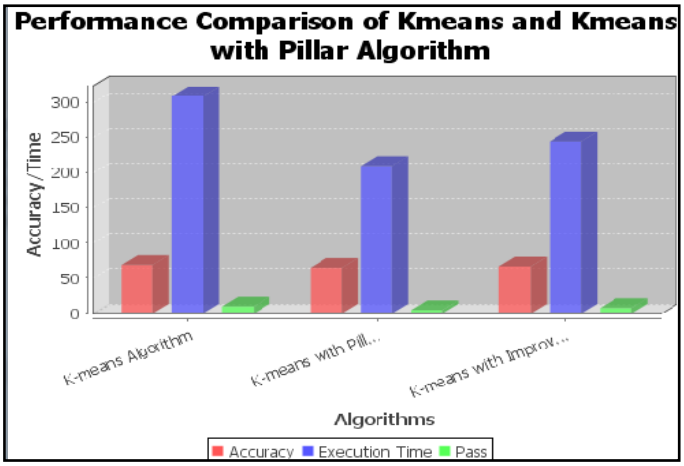
a. Iris dataset



b. Thyroid dataset

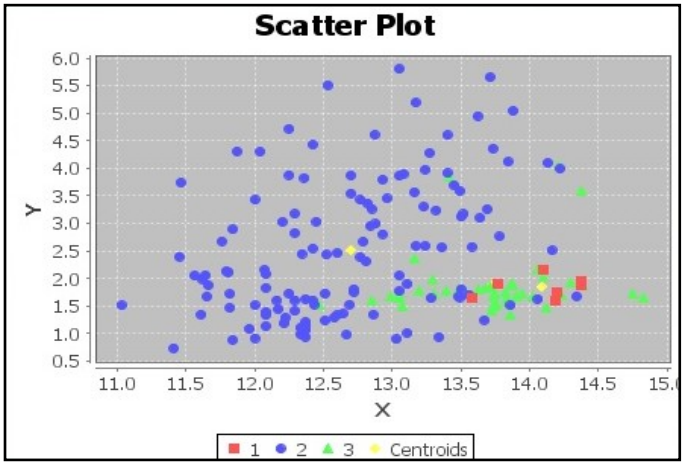


b. Thyroid dataset



c. Wine dataset

Fig.5. Comparison of proposed approach and K Means random initialization with accuracy, time and iterations for different datasets.



c. Wine dataset

Fig.6. Clusters Scatter Plot for different datasets.

VI. CONCLUSION

The K means clustering algorithm which mainly suffers from initial cluster centers. The main objective of the proposed system is to optimize the K means clustering algorithm by designating the initial centroids using Pillar algorithm. This spreads them in the feature space so that distance among them is as far as possible. Pillar algorithm is very effective to position the initial centroids and improve the accuracy of clustering results. However, inappropriate parameter set up in the proposed algorithm for outlier detection may lead to reduced performance. Adjusting to the characteristics of data distribution in data set is needed in order to set up the appropriate parameters for outlier detection mechanism.

An improvement is done in pillar algorithm by reducing number of distance calculation of the previous initial centroids neighbors for next step of iterations to speed up the computational time. The reduced complexity of Pillar algorithm from O ((k+h+1) n) to O (n + (h_1. n_1) +…+ (h_k. n_k)) in which number of involved data for distance calculation in next steps are reduced. The experimental results show the improved solution using the proposed approach.

REFERENCES

[1] Y. M. Cheung, "K*-Means: A new generalized k-means clustering algorithm" Pattern Recognition Lett., 24, 2883-2893, 2003.

[2] S. S. Khan, A. Ahmad, "Clustering center initialization algorithm for K-means clustering," Pattern Recognition Lett., 25, 1293-1302, 2004.

[3] A. R. Barakbah, K. Arai, "Hierarchical K-means: an algorithm for centroids initialization for K-means," Report of the Faculty of Science and Engineering, Saga University, Japan, Vol. 36, No. 1, 2007.

[4] P. S. Bradley, U. M. Fayyad, "Refining initial points for K-means clustering," proc. 15th International Conference on Machine Learning (ICML 98), 1998.

[5] J. M. Pena, J. A. Lozano, P. Larranaga, "An empirical comparison of the initialization methods for the K-means algorithm," Pattern Recognition Lett., 20, 1027-1040, 1999.

[6] A. R. Barakbah, A. Helen, "Optimized K-means: an algorithm of initial centroids optimization for K-means," Proc. Soft Computing, Intelligent System, and Information Technology (SIIT) 2005, pp.2-63-66, Petra Christian University, Surabaya, 2005.

[7] A. R. Barakbah, "A new algorithm for optimization of K-means clustering with determining maximum distance between centroids," Proc. Industrial Electronics Seminar (IES) 2006, pp.240-244, Electronic Engineering Polytechnic Institute of Surabaya-ITS, Surabaya, 2006.

[8] A. R. Barakbah, Y. Kiyoki, "A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation," The IEEE Symposium on Computational Intelligence and Data Mining, Nashville, 2009.

[9] UCI Repository (http://www.sgi.com/tech/mcl/db/)

