Outliers Elimination: A Modified Clustering Technique of K-means Algorithm

Md.Mahfuz Reza, Tanoy Debnath, Sadee Ibn Sultan

Abstract— Data Mining refers to extracting or mining knowledge from a huge amount of data. Clustering is an important data analytic technique which has a significant role in data mining application. Clustering is the method of arranging a set of similar objects into a group. A widely used partition based clustering algorithm is k- means clustering. Among various types of clustering techniques, K-Means is one of the most popular algorithms. The objective of K-means algorithm is to make the distances of objects in the same cluster as small as possible. But this algorithm also has some limitations. These limitations can be reduced by some improvements in existing algorithm. This paper has been proposed a modified technique of K-Means clustering approach which is better in the process in large number of clusters. The proposed algorithm can give better result whether dataset contain outlier and also can handle large number of data set and produce result with a minimum period of time compare to the other clustering technique.

Index Terms— Data Mining, Clustering, K-means Clustering, Weighted Average

1 INTRODUCTION

Clustering is a technique of grouping data objects into a object so that the data in the same cluster are similar. A cluster is collections of data that are similar to one another are in same cluster and dissimilar to the objects are in other clusters. The demand for organizing the sharp increasing data and learning valuable information from data, which makes clustering techniques are widely applied in many application areas such as Neural Network, Biology, Data Mining, Image Processing, Machine Learning, Pattern Recognition, Psychology, Telecommunication, Banking Industry, Statistics and so on. Clustering is unsupervised learning and do not rely on predefined classes. In clustering we measure the dissimilarity between objects by measuring the distance between each pair of objects. These measure include the Euclidean, Manhattan and Minkowski distance.

2 RELATED WORKS

In order to get efficient and effective result of K-mean algorithm there have been a lot of research happened in previous day. All researchers worked on different view and with different idea[1]. Krishna and Murty proposed the genetic K-means(GKA) algorithm which integrate a genetic algorithm with K-means in order to achieve a global search and fast convergence. Jain and Dubes recommend running the algorithm several times with random initial partitions. The clustering results on these different runs provide some insights into the quality of the ultimate clusters [4]. Forgy's method generates the initial partition by first randomly selecting K points as prototypes and then separating the remaining points based on their distance from these seeds[2]. Likas proposed a global K-means algorithm consisting of series of K-means clustering procedures with the number of clusters varying from 1 to K. One disadvantage of the algorithm lies in the requirement for executing K-means N times for each value of K, which causes high computational burden for large data sets.

3 K-MEANS ALGORITHM

K-means is the simplest and most popular classification clustering method that is easy to implement. The classification method can only be used if the data about all the objects is located in the main memory [5]. The method is called K-means since each of the K clusters is represented by the mean of the objects (called the centroid) within it.It is also called the centroid method since at each step the centroid point of each cluster is assumed to be known and each of the remaining points are allocated to the cluster whose centroid is closed to it. Once this allocation is completed, the centroids of the clusters are recomputed using simple means and the process of allocating points to each cluster is repeated until there is no change in the clusters.

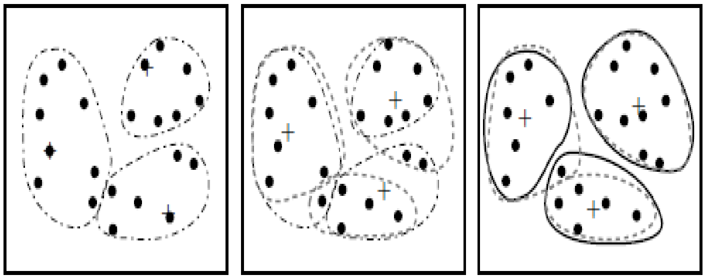


Figure: Clustering of a set of objects based on the *k*-means method. (The mean of each cluster is marked by a "+".)

The K-means method uses the Euclidean distance measure, which appears to work well with compact clusters. If instead of the Euclidean distance, the Manhattan distance is used the method is called the K-median method. The K-median method can be less sensitive to outliers.

Algorithm: *k*-means.

The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:
k: the number of clusters,
D: a data set containing *n* objects.
Output: A set of *k* clusters.
Method:

1. Select the number of clusters. Let this number be k.

2. Pick k seeds as centroids of the k clusters. The seeds may be picked randomly unless the user has some insight into the data.

3. Compute the Euclidean distance of each object in the dataset from each of the centroids.

$$d(x_i\, y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

4. Allocate each object to the cluster it is nearest to based on the distances computed in the previous step.

5. Compute the centroids of the clusters by computing the means of the attribute values of the objects in each cluster.

6. Check if the stopping criteria has been met(the cluster membership is unchanged). If yes, go to Step 7. If not, go to Step 3.

7. One may decide to stop at this stage or to split a cluster or combine two clusters heuristically until a stopping criterion is met.

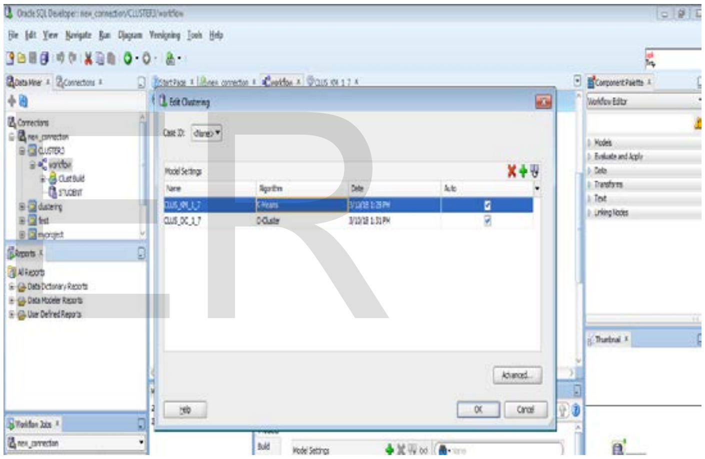
4 PROPOSED METHOD

Input: Dataset of N data points D (i = 1 to N)
Desired number of clusters = k
Output: N data points clustered into **k clusters.**
Steps:

```
    1. Input the data set and value of k.
    2. If the value of k is 1 then Exit.
    3. Else
{
  a) Calculate the average value of each data point.
    1. di= x1, x2,x3,x4…xn
    2. di(avg)=(w1*x1+w2*x2+w3*x3+…..wm*xm)/m where, x= attribute's value , m= no of attributes, w= weight to multiply to ensure fair distribution of cluster.
  b) Sort the data based on average value. .
  c) Divide the data based on k subsets.
  d) Calculate the mean value of each subset.
  e) Take the nearest possible data point of the mean as the initial centroid for each data subsets.
}
    4. After initialize the centroid ,
    5. do the following steps :----
    6. Compute the distance of each data-point di (1<=i<=n) to all the centroids cj(1<=j<=k) as d(di, cj);
    7. For each data-point di, find the closest centroid cj and Assign di to cluster j.
    8. Set Cluster Id[i]=j and Set Nearest _Dist[i]=d(di, cj); // j: Id of the closest cluster
    9. For each cluster j (1<=j<=k), Recalculate the centroids;
  10. Repeat
  11. For each data-point di, Compute its distance from the centroid of the present nearest cluster;
      a) If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster;
      b) Else for every centroid cj(1<=j<=k) compute the distanced(di, cj);
    End for;
  12. Assign the data-point di to the cluster with the nearest centroid cj
  13. Set ClusterId[i]=j and Set Nearest_Dist[i] = d(di, cj);
    End for (step (7));
  14. For each cluster j (1<=j<=k), Recalculate the centroids until the convergence criteria is met.
```

5 EXPERIMENTAL RESULTS AND ANALYSIS

We evaluate our algorithm using randomly generated dataset and compare with various proposed methods namely Standard K-means Algorithm, Improve k-means Clustering Algorithm by Nazeer and Sebastian , Optimized Version of the K-Means Clustering Algorithm by Poteras , Miha˘escu and Mocanu. We use Oracle Data Miner (ODM) to simulate the result[1].

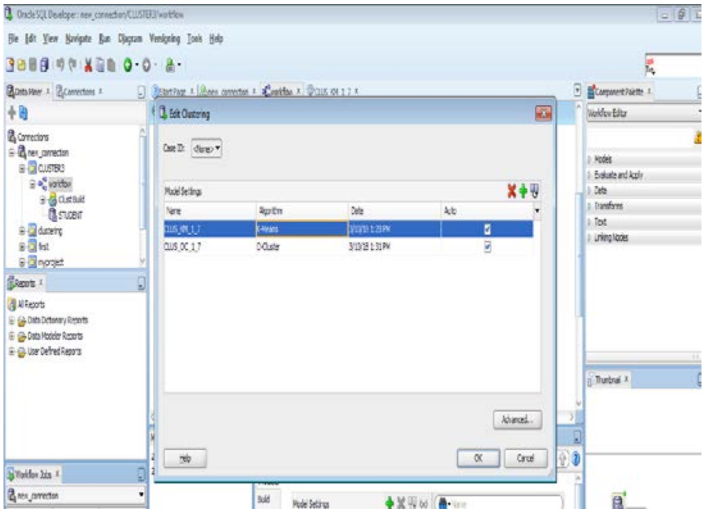


Edit Clustering

Case ID: <None>

Model Settings

Name	Algorithm	Date	Auto
CLUS_KM_1_7	K-Means	3/10/18 1:29 PM	☑
CLUS_OC_1_7	O-Cluster	3/10/18 1:31 PM	☑



Edit Clustering

Case ID: <None>

Model Settings

Name	Algorithm	Date	Auto
CLUS_KM_1_7	K-Means	3/10/18 1:29 PM	☑
CLUS_OC_1_7	O-Cluster	3/10/18 1:31 PM	☑

Figure 1: K-means clustering in ODM
Figure 2: Centroids Selection in ODM

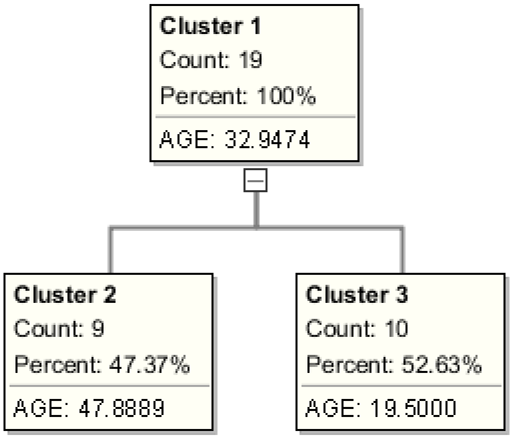


Figure 3: Final Result

The concept is described by the following example:
Suppose n=Number of data points=19
K=Number of cluster=2
The given datasets Di
(15,15,16,19,19,20,20,21,22,28, 35,40,41,42,43,44,60,61,65)

K-means:
Initial Cluster:
Centroid C1=16
Centroid C2=22

Iteration 1:
C1=15.35(15,15,16)
C2=36.25(19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65)

Iteration 2:
C1=18.56(15,15,16,19,19,20,20,21,22)
C2=45.90(28,35,40,41,42,43,44,60,61,65)

Iteration 3:
C1=19.50(15,15,16,19,19,20,20,21,22,28)
C2=47.89(35,40,41,42,43,44,60,61,65)

Iteration 4:
C1=19.50(15,15,16,19,19,20,20,21,22,28)
C2=47.89(35,40,41,42,43,44,60,61,65)

Proposed Approach:
Initial Centroid= After normalize and sorting the datasets, the total data points can be divided into k subsets.
C1=(15+15+16+19+19+20+20+21+22)/9=18.56
C2=(28+35+40+41+42+43+44+60+61+65)/10=45.9
After initializing the centroids
C1=18.56
C2=45.9

Iteration 1:
C1=19.50(15,15,16,19,19,20,20,21,22,28)
C2=47.89(35,40,41,42,43,44,60,61,65)

After sorting the data, the proposed method only take a single iteration to produce the result.If there is big value most of the case all the algorithm failed to cluster the dataset within short period of time but our proposed method will produce the result with small number of iterations.

The following tables illustrate the comparison of the proposed methods with the existing methods.

Algorithm	Data Points	Number of Clusters	Initial Cluster
K-means	Input by User	Input by User	Randomly Chosen by User
Improve K-means	Input by User	Input by User	Calculated Value
Optimized K-means	Input by User	Input by User	Randomly Chosen by User
Proposed Algorithm	Input by User	Computed by Algorithm	Computed by Algorithm

Table 1: Algortihm Training

Algorithm	Initial Centroids	Accuracy (%)	Time Taken (ms)	Complexity
K-means Algorithm	5.1,3.5,1.4,0.2 4.3,3,1.1,0.1 6.6,2.9,4.6,1.3	52.6	71	*O(n^2).*
(Executed 7 times with randomly selected initial centroid)	7,3.2,4.7,1.4 6.7,3.1,4.4,1.4 5.1,3.5,1.4,0.2	88.7	69	
	7,3.2,4.7,1.4 6.7,3.1,4.4,1.4 7.4,2.8,6.1,1.9	89.3	70	
	7.4,2.8,6.1,1.9 6,3,4.8,1.8 6.7,3.1,4.4,1.4	89.3	72	
	5.1,3.5,1.4,0.2 4.3,3,1.1,0.1 6,3,4.8,1.8	52.7	70	
	6,3,4.8,1.8 5.8,2.7,5.1,1.9 5.1,3.5,1.4,0.2	89.3	72	
	5.1,3.5,1.4,0.2 7,3.2,4.7,1.4 6.3,3.3,6,2.5	89.3	71	
Mean value	-----------------------	78.7	70.7	
Proposed Algorithm	Computed by the Algorithm	**88.6**	**67**	***O(nk)***

Table 2: Experimental result

5.1 Complexity Analysis

Complexity can be categorized into two ways. Time complexity and space complexity. Phase I of the enhanced algorithm requires a time complexity of $O(n^2)$.For finding the initial centroids, as the maximum time required here is for computing the distances between each data point and all other data-points in the set D. In the original k-means algorithm, before the algorithm converges the centroids are calculated many times and the data points are assigned to their nearest centroids. Since complete redistribution of the data points takes place according to the new centroids, this takes $O(nkl)$, where n is the number of data-points, k is the number of clusters and l is the number of iterations. To obtain the initial clusters, Algorithm requires $O(nk)$. Here, some data points remain in its cluster while the others move to other clusters depending on their relative distance from the new centroid and the old centroid. This requires $O(1)$ if a data-point stays in its cluster, and $O(k)$ otherwise. Hence the total cost of this phase of the algorithm is $O(nk)$, not $O(nkl)$. Thus the overall time complexity of the enhanced algorithm becomes $O(n^2)$, since k is much less than n.Total time required by improved algorithm is o(nk) while total time required by standard k-mean algorithm is o(nkt). So the improved algorithm improve clustering speed and reduce the time complexity.

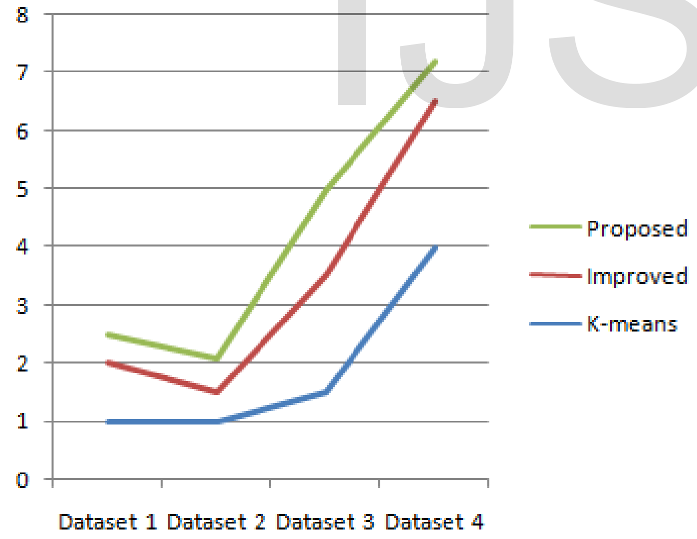


Figure 4: Complexity analysis of different algorithms.

6 Conclusion and Future Work

The K-means clustering algorithm is widely used for clustering huge amount of data. But traditional k means algorithm does not always produce good result. This paper presents an efficient algorithm where the initial cluster is computed by the algorithm so that outliers can be detected and can't slow down the process and generates effective clusters. The proposed algorithm has proved to be better than traditional K-means algorithm in terms of execution time.

References

1. SK Ahammad Fahad, Md. Mahbub Alam "A Modified K-Means Algorithm for Big Data Clustering"- International Journal of Computer Science and Engineering Technology, Vol 6, Issue 4, 129-132.
2. Shailendra Singh Raghuwanshi, PremNarayan Arya"Comparison of K-means and Modified K-means algorithms for large Data-set"- International Journal of Computing, Communications and Networking, Volume 1, No.3, November – December 2012
3. Deepali Virmani,Shweta Taneja,Geetika Malhotra "Normalization based K-means Clustering Algorithm",arxiv.org
4. Ahamed Shafeeq B M and Hareesha K S "Dynamic Clustering of Data with Modified K-Means Algorithm" *International Conference on Information and Computer Networks (ICICN 2012),IPCSIT vol. 27 (2012)*
5. Anil K. Jain and Richard C. Dubes, Michigan State University; *Algorithms for Clustering Data:* Prentice Hall, Englewood Cliffs, New Jersey 07632. ISBN: 0-13-0222278-X.
6. Ritu Yadav & Anuradha Sharma" Advanced Methods to Improve Performance of K-Means Algorithm: A Review", Global Journal of Computer Science and Technology Volume 12 Issue 9 Version 1.0 April 2012
7. Anshul Yadav, Sakshi Dhingra "An Enhanced K-Means Clustering Algorithm to Remove Empty Clusters", International Journal of Engineering Development and Research, Volume 4, Issue 4 | ISSN: 2321-9939
8. Forgy E (1965) *Cluster analysis of multivariate data; efficiency vs. interpretability of classifications.* Biometrics, 21: pp 768-780
9. Bradley P, Fayyad U (1998) *Refining initial points for K-means clustering.* International conference on machine learning (ICML- 98), pp 91-99
10. Krishna K, Murty M (1999) *Generic K-Means algorithm.* IEEE Transactions on systems, man, and cybernetics-part B: Cybernetics, 29(3): pp 433-439
11. Likas A, Vlassis N, Verbeek J (2003) *The global K-means clustering algorithm*. Pattern recognition, 36(2), pp 451-461
12. Pena JM, Lozano JA, Larranaga P (1999) *An empirical comparison of four initialization methods for K-means algorithm.* Pattern recognition letters 20: pp 1027-1040
13. Ball G, Hall D (1967) *A clustering technique for summarizing multivariate data.* Behavioral science, 12: pp 153-155
14. Milligan G, Cooper M (1985) *An examination of procedures for determining the number of clusters in a data set.* Psychometrika, 50: pp 150-179
15. SAS Institute Inc., *SAS technical report A-108 (1983) Cubic clustering criterion.* Cary, NC: SAS Institute Inc., 56 pp
16. CosminMarianPoteras,MarianCristianMiha˘escu ,MihaiMocanu *An Optimized Version of the K-Means*

*Clustering Algorithm.*University of Craiova. Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 695–699.

17. K. A. Abdul Nazeer, M. P. Sebastian *Improving the Accuracy and Efficiency of the k-means Clustering Algorithm.* WCE 2009, July 1- 3, 2009, London, U.K.
18. Rui X, Wunsch DC II (2009*) Clustering*. IEEE Press series on computational intelligence, John Wiley & Sons.
19. Jian Zhu, Hanshi Wang "An improved K-means Clustering Algorithm" 2010 IEEE.
20. S. Prakash kumar and K. S. Ramaswami, "Efficient Cluster Validation with K-Family Clusters on Quality Assessment", European Journal of Scientific Research, 2011, pp.25-36.