"Bodh", BPIT's International Journal of Technology & Management

ISSN: 2454-8421, Volume 1, Issue 1, July-Dec, 2015. *Page 70-74*

A NEWAPPROACH TOWARDS K-MEANS ALGORITHM USING SEGMENTATION

Preeti Arora
Assistant Professor,
CSE Dept
BPIT, Delhi
erpreetiarora07@gmail.com

Pooja Mudgil
Assistant Professor,
IT Dept
BPIT, Delhi
engineer.pooja@gmail.com

Shipra Varshney
Assistant Professor,
CSE Dept
NIEC, Delhi
shipra_vin@yahoo.com

Abstract*— *Nowadays data mining is used in many fields for the extraction of similar information from the large data volumes. The data before information extraction contains noise which is then removed such that the predictive information can be extracted. The predictive information so produced helps in the business analysis of an organization. Clustering is one of the techniques applied for knowledge discovery to group the data on the basis of similarities and dissimilarities among the data elements and generally for this purpose K-Means Algorithm is applied. In this paper, a new data clustering approach called enhanced K-Means algorithm is proposed where improvement is made on the initial selection of centroids for the clusters. The centroids are chosen such that the whole space is divided into different segments of precise range and then calculates the frequency of data points in each segment thereby assigns the data point to their appropriate cluster. This process works more efficiently as it reduces the time complexity, the effort of numerical calculation and retains the easiness of implementing the K-mean algorithm.

Keywords— ***K-means, data clustering, centroid, segment***

I. Introduction

A basic problem that frequently arises in different fields like data mining and knowledge discovery [1], data compression and vector quantization [2], and pattern recognition and pattern classification [3] is the clustering problem. It also has been applied in a great variety of applications, such as image segmentation, document retrieval, object and character recognition [4]. The importance of data mining is rising exponentially since last decade. There is a large amount of data available in real world which makes it very difficult to access the useful information from this vast database and provide the information which is required within time limit and in required outline. So data mining provides the way to remove the noise from data and extract information from large database and give it in the form in which it is required for each specific job. The use of data mining is very immense in today's scenario [5].

Cluster analysis of data is widely used in knowledge discovery and data mining. It aims to group data on the basis of similarities and dissimilarities among the data elements so that we have high intra class similarity and low inter class similarity and can be performed in a supervised or unsupervised way.

II. Literature Review

Although the work has been done by various authors on the initial selection of cluster centroids in which centroid selection is an independent initialization, to optimize the clustering approach? The most notable work has been briefly discussed in this section.

In paper [5] author defined a threshold distance for each cluster's centroid to compare the distance between data point and cluster's centroid with this threshold distance through which they could reduce the computational effort while the calculation of distance between data point and cluster's centroid. It is shown that how the modified K-mean algorithm will lessen the complexity & the effort of numerical calculation, preserving the easiness of implementing the K-mean algorithm. It assigns the data point to their appropriate class or cluster more efficiently.

In paper [6] author define a modified K-mean algorithm in which it has been discussed about the limitations of K-mean algorithm and improvement has been done to increase the speed and efficiency of K-mean algorithm. Their algorithm removes the need of specifying the value of K in advance which is practically very difficult. Our proposed algorithm is better in two ways as compared to the others as discussed above. First, it results in optimal number of cluster and second it reduces computational complexity and remove dead unit problem.

III. PROPOSED ALGORITHM

The K-means algorithm is a well-known partitioning method for clustering. K-means clustering method, groups the data based on their closeness to each other accordin; to the Euclidean distance. In this clustering approach the user decides that how many cluster should be, but the clusters are incremented dynamically in phase 1. For each data vector this algorithm calculate the distance between data vector and each cluster centroid using equation (1).

$$D(Zp, Mj) = \sqrt{\sum(Zp,k - Mj,k)} \quad \text{............(1)}$$

Z_p is p^{th} data point
M_j is centroid of j^{th} cluster.

The centroid is recalculated each time respectively after addition of data point in cluster j. It is calculated using equation (2)

$$M_j = 1/N_j \sum Zp, \forall Zp \in Cj \quad \text{...........(2)}$$

where N_j is the number of data point in cluster j.

The present work has overcome the limitations that were in the paper [5]. Enhancement has been done in modified K-mean algorithm by dividing the whole space is divided into different segments of precise range. The segment which shows maximum frequency will have the highest probability to have the centroid of the cluster. The number of cluster's centroid (K) will be provided by the user in the same way like the traditional K-mean algorithm but will be dynamically increased under some conditions and the number of division will be k*k ('k' vertically as well as 'k' horizontally). If the highest frequency of data point is same in different segments and the upper bound of segment exceeds the threshold 'k' then merging of different segments become compulsory and then take the highest k segment for calculating the initial centroid of clusters. In this paper we define a threshold distance for each cluster's centroid in which we compare the distance between data point and cluster's centroid with this distance by which we can lessen the computational effort. Although, after addition of data point to the cluster the centroid is recalculated by taking mean of all data points in that cluster.

As we know that K-mean is widely used in many areas because of its simplicity and easiness to implement. It requires less computation but there are some limitations:-

1. Initial selection of the number of cluster should be previously known and specified by the user.
2. Results directly depend on the initial centroid of cluster.
3. It can contain the dead unit problem.

Our proposed work will provide the solution for the above limitations. The first limitation can be minimized by running the algorithm for different number of K- values and increasing them dynamically after analyzing the density of data points. The proposed algorithm is based on density of different regions which eventually minimizes 2^{nd} limitation and hence will solve the problem of dead unit point because the centroid of cluster is located in the first iteration pertaining to the maximum density of the data points. In this approach data points are taken from UCI dataset. After taking the data set as input, user defines the 'k' value, where 'k' denotes the number of clusters. Suppose the value of k defined by user is 4, this means user has defined 4 clusters. Then the space will be partitioned into k*k segments.

Phase 1:

In this approach 67 data points are taken and subsequently plotted as in Fig 1. After taking the data set as input, user defines the 'k' value, where 'k' denotes the number of clusters.

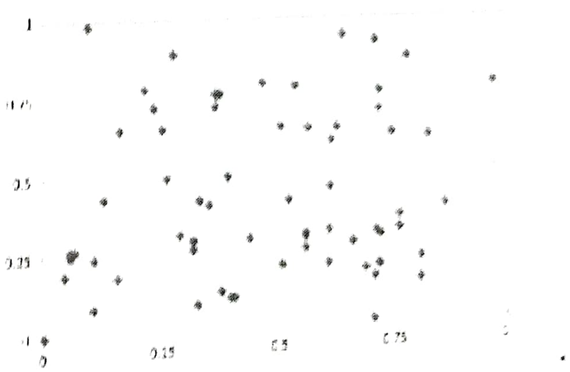

Chart 1: Data Set

Suppose the value of k defined by user is 4, i.e. user defines 4 clusters. Then the space will be divided into k*k segments, as shown in fig 2.

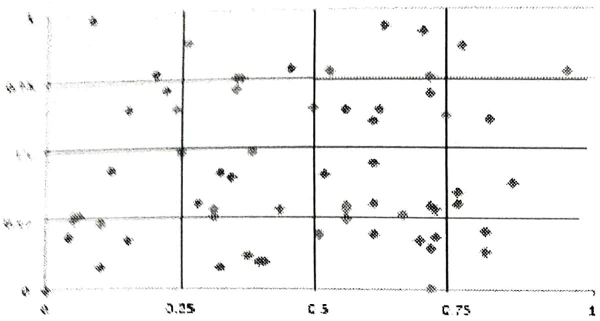

Chart 2: XY plane partitioned into different segments

Segment(rectangle)	No. of data points (frequency)
(0,0)-(.25, .25)	11
(.25,0)-(0.5,0.25)	0
(.5,0)-(0.75,0.25)	7
(.75,0)-(1,0.25)	6
(0,0.25)-(0.25,0.50)	6
(0.25,0.25)-(0.5,0.5)	8
(0.5,0.25)-(0.75,0.5)	2
(0.75,0.25)-(1,0.5)	3
(0,0.5)-(0.25,0.75)	3
(0.25,0.5)-(0.5,0.75)	5
(0.5,0.5)-(0.75,0.75)	6
(0.75,0.5)-(1,0.75)	10
(0,0.5)-(0.25,1)	4
(0.25,0.75)-(0.5,1)	1
(0.5,0.75)-(0.75,1)	3
(0.75,0.75)-(1,1)	4

Table 1: Group Frequencies

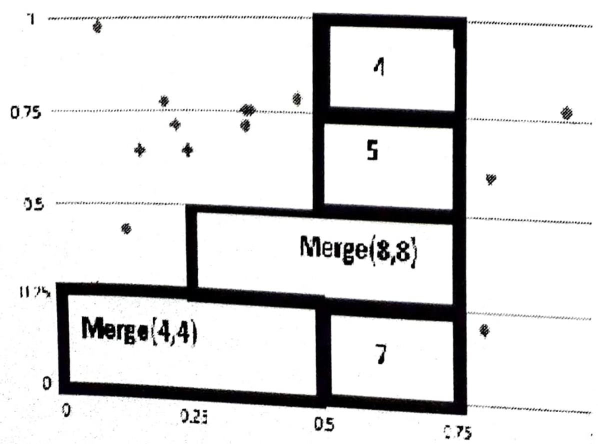

Fig 1: Segments with highest frequencies

The adjacent segments with the same frequencies are merged into one segment. Then the mean of all data points is taken which are coming in that segment. If the segments with same frequency are not adjacent, then a new cluster is generated. This makes the clusters dynamic. Thus, initial centroids are calculated.

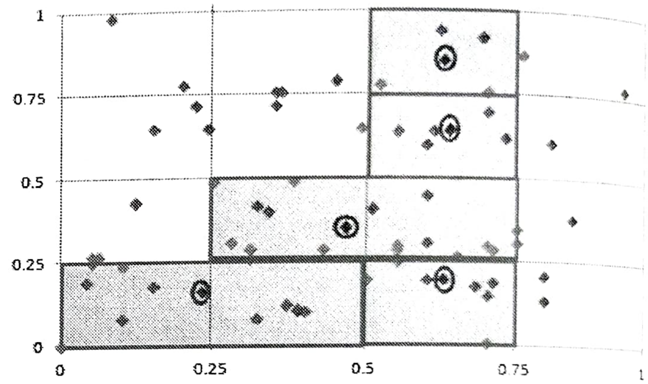

Fig 2: Initial centroids

Phase 2:

To assign the data point to appropriate cluster's centroid, we calculate the distance between each cluster's centroid and for each centroid take the minimum distance from the remaining centroid and make it half, denoted by DC (i) i.e. half of the minimum distance from ith cluster's centroid to the other cluster's centroid. Now take any data point to calculate its distance from ith centroid and compare it with DC(i). If it is less than or equal to DC (i) then data point is allocated to the ith cluster or else calculate the distance from the other centroid. Repeat this process until that data point is allocated to any of the remaining cluster. After assigning the data point to that cluster, mean is calculated, and centroid keeps on moving in contrast to previous algorithm where centroid was calculated after the complete iteration. If data point is not assigned to any of the cluster then the centroid which shows the minimum distance with data point becomes the cluster for that data point. Repeat this process for each data point. Repeat phase 2 until termination condition is achieved.

No: Number of data points

K: Number of clusters' centroids

C_i: i^{th} cluster

Equations used in algorithm are:

$|C_i, C_j| = \{d\,(m_i, m_j) : (i,j) \in [1,k]\ \&\ i \neq j\}$…….(3)

Where $|C_i, C_j|$ is the distance between cluster C_i and C_j

$DC_{(i)} = 1/2(\min \{|C_i, C_j|\}$ ……………………..(4)

Where $DC_{(i)}$ is half of the minimum distance from i^{th} cluster to any other remaining cluster.

1. Input the data set and value of k.
2. If the value of k is 1 then Exit.
3. Else
4. /*divide the data point space into k*k, means k vertically and k horizontally*/
5. For each dimension {
6. Calculate the minimum and maximum value of data points
7. Calculate range of group (R_G) using equation ((min+max)/k)
8. Divide the data point space in k group with width RG
9. }
10. Calculate the frequency of data points in each partitioned space.
11. Choose the k highest frequency group.
12. If same frequency segments are adjacent
13. Merge the segments
14. Go to step 17
15. Else
16. k=k+1(Make new cluster)
17. Calculate the mean of selected group. /* This will be the initial centroid of cluster.*/
18. Calculate the distance between each clusters using equation (3)
19. Take the minimum distance for each cluster and make it half using equation (4)
20. For each data points p= 1 to No {
21. For each cluster j= 1 to k {
22. Count the number of data points in Cj
23. if (count==1)
24. Calculate d(Zp,Mj) using equation (1)
25. If (d(Zp,Mj) < DCj){
26. Then Zp assign to cluster Cj
27. Break
28. }
29. Else if
30. Take the mean of all data points in Cj
31. Go to step 24
32. Else
33. Continue;
34. }
35. If Zp does not belong to any cluster then
36. Zp $\in$ min (d(Zp , Mi)) where i$\in$ [1, Nc]
37. }
38. Check the termination condition of algorithm if satisfied
39. Exit.
40. Else
41. Go to step 13.

In the above algorithm steps 5-17 is one time execution step and it ensures the non existence of dead unit problem and optimizes the selection of initial centroid of cluster by using the most densely populated area as the centroid of cluster. This takes unit time for execution, so elapsed time will not increase rather it will decrease because initial centroid location is improved. As a result, number of iterations will decrease therefore overall execution time will decrease. Steps 12 to 16 define a new cluster whenever same frequency segments are not adjacent. Steps 13 to 27 ensure the minimum execution time during the allocation of data points to respective cluster because each time the modified algorithm tests from threshold. This ensures that outliers will be minimised. Also when number of cluster increases manifold the modified algorithm will take less time compared to the traditional algorithm because traditional algorithm calculates distance from data points with each cluster wasting significant amount of time. Step 30 calculates mean of data points in the specified clusters, this reduces the number of iterations. Thus, making convergence criteria achieve easily. In our approach it is not required to calculate the distance from data point to each cluster rather in best case we are required to calculate distance for each data point to only one cluster therefore increase in the number of clusters would prove to be more significant. Also in average case the elapsed time will be less than the traditional k-means algorithm for the same reason.

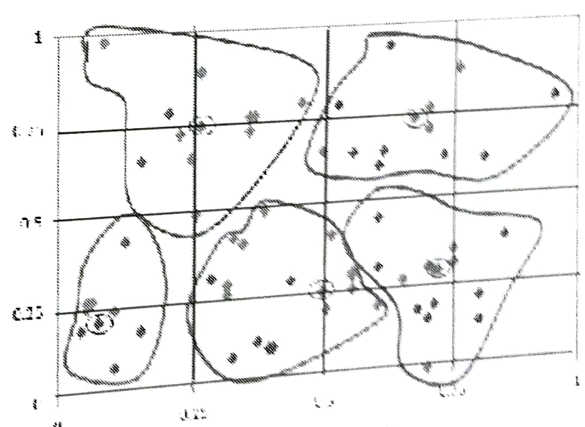

Fig 3: Final Result

IV. Conclusion

Data clustering is a process of keeping similar data together which means similarity among data within the cluster will be maximum and among the clusters would be minimum. K-Means is a very important method for data clustering. We have defined an improved version of this K-Means which increases the number of clusters dynamically according to the density of data points and it does not depend on the ordering of data. The computational efforts are minimized by incorporating the threshold value and calculating the mean of all data points in the cluster at each step, thus, minimizing the occurrence of outliers.

REFERENCES

[1]. D. Napolean , P . Ganga lakshmi , "An Efficient K-Means Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points ", *Proceedings of the 2010 Conference on Trendz in Information Sciences and Computing* (TISC'10) , 17-19 Dec, 2010.

[2]. Shi Na, Liu Xumin and Guan yong , "Research on means Clustering Algorithm: An Improved k-means Clustering Algorithm" *Proceedings of the 2010 Conference on Intelligent Information Technology and Security Informatics* (IITSI' 10), Third International Symposium on 2-4 April, 2010.

[3]. Ran Vijay Singh, M.P.S. Bhatia , " Data Clustering with modified K-means algorithm ", *Proceedings of the 2011 International Conference on Recent Trends in Information Technology (ICRTIT'11),* 3-5 June 2011.

[4]. JYOTI YADAV , MONIKA SHARMA , " A REVIEW OF K-MEANS ALGORITHM " , *PROCEESSDINGS OF THE INTERNATIONAL JOURNAL OF ENGINEERING TRENDS AND TECHNOLOGY (IJETT'13) – VOLUME 4 ISSUE* 7-JULY, 2013.

[5]. TEKNOMO, KARDI , "K-MEANS CLUSTERING TUTORIALS",*HTTP:\\PEOPLE.REVOLEDU.COM\KARDI\TUTORIAL\KMEAN* , JULY 2007.

[6]. Tapas Kanungo , Nathan S. Netanyahu and Angela Y. Wu, "*An Efficient k-Means* and *Clustering Algorithm: Analysis and Implementation*", IEEE transactions on pattern analysis machine intelligence, vol. 24, no. 7, July 2002

[7]. Mingwei Leng, Haitao Tang and Xiaoyun Chen, "An Efficient Kmeans Clustering Algorithm Based on Influence Factors" *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.*

[8]. Baolin Yi, Haiquan Qiao, Fan Yang, Ch enwei Xu, "An Improved Initialization Center Algorithm for K-means Clustering," IEEE 2010 .

[9]. Mingwei Leng, Haitao Tang and Xiaoyun Chen, "An Efficient Kmeans Clustering Algorithm Based on Influence Factors" *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.*

[10]. Baolin Yi, Haiquan Qiao, Fan Yang, Chenwei Xu, "An Improved Initialization Center Algorithm for K-means Clustering," *IEEE 2010* .

[11]. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press (1996)

[12]. A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic, Boston, MA(1992)

[13] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York(1973)

[14]. M.N. Murty, A.K. Jain, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31(3) (1999) 264-323.