A Review of K-mean Algorithm

Jyoti Yadav[#1], Monika Sharma[*2]

[1]PG Student, CSE Department, M.D.U Rohtak, Haryana, India

[2] Assistant Professor, IT Department, M.D.U Rohtak, Haryana, India

***Abstract*— Cluster analysis is a descriptive task that seek to identify homogenous group of object and it is also one of the main analytical method in data mining. K-mean is the most popular partitional clustering method. In this paper we discuss standard k-mean algorithm and analyze the shortcoming of k-mean algorithm. In this paper three dissimilar modified k-mean algorithm are discussed which remove the limitation of k-mean algorithm and improve the speed and efficiency of k-mean algorithm. First algorithm remove the requirement of specifying the value of k in advance practically which is very difficult. This algorithm result in optimal number of cluster Second algorithm reduce computational complexity and remove dead unit problem. It select the most populated area as cluster center. Third algorithm use simple data structure that can be used to store information in each iteration and that information can be used in next iteration. It increase the speed of clustering and reduce time complexity.**

***Keywords* — Clustering, K-mean, Computational Complexity**

I. INTRODUCTION

Clustering is a process of grouping data objects into disjointed clusters so that the data in the same cluster are similar, but data belonging to different cluster differ. A cluster is a collection of data object 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 artificial intelligence, biology, customer relationship management, data compression, data mining, information retrieval, image processing, machine learning, marketing, medicine, pattern recognition, psychology, statistics and so on. Cluster analysis is a tool that is used to observe the characteristics of cluster and to focus on a particular cluster for further analysis. 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.

Euclidean distance is defined as

$$d(i,j) = \sqrt{\left(\sum_{i=0}^{n}(x_i - y_i)^2\right)}$$

Where

i = x_i and j = y_i are two n-dimensional data objects.

Manhattan distance defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots\ldots + |x_{in} - x_{jn}|$$

Minkowski distance is a generalization of both Euclidean distance and Manhattan distance. It is defined as

$$d(i, j) = (|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \ldots\ldots + |x_{in} - x_{jn}|^p)^{1/p}$$

Where

p is a positive integer. Such a distance is called L_p norm. It represents the Manhattan distance when p=1 and Euclidean distance when p=2.

K-mean

K-mean is the most popular partitioning method of clustering. It was firstly proposed by *MacQueen* in 1967. K-mean is a unsupervised, non-deterministic, numerical, iterative method of clustering. In k-mean each cluster is represented by the mean value of objects in the cluster. Here we partition a set of n object into k cluster so that intercluster similarity is low and intracluster similarity is high. Similarity is measured in term of mean value of objects in a cluster .

The algorithm consists of two separate phases.

1st Phase

select k centroid randomly, where the value k is fixed in advance.

2nd Phase

Each object in data set is associated to the nearest centroid.

Euclidean distance is used to measure the distance between each data object and cluster centroid.

K-mean Algorithm

Input

K : number of desired cluster
D : {d1, d2,…..dn} a data set containing n objects.

Output

A set of k cluster as specified in input

Method

1) Arbitrarily choose k data item from D dataset as initial cluster centriod;
2) Repeat
3) Assign each data item di to the cluster to which object is most similar based on the mean value of the object in cluster;
4) Calculate the new mean value of the data items for each cluster and update the mean value;
5) Until no change.

Firstly we arbitrarily select the value of k from data set D which specify the desired number of cluster. Each value of k initially represent the center of cluster or mean of cluster. For each of the remaining data item in D, data item is assigned to the cluster to which it is the most similar, based on distance between data item and cluster mean. Then compute the new mean for each cluster and update the mean value. Repeat this process until centroid tends to be unchangeable or we can say that until the criterion function converges.

In k-mean algorithm the intercluster distance i.e the minimum distance between cluster center is measured as
Inter =min { $m_k - m_{kk}$ } for all k = 1,2……k-1 and kk = k+1……..k ……..(1)

Drawback of K-mean Algorithm

1) Sensitive to the selection of initial cluster center.
2) There is no rule for the decision of value of k and sensitive to initial value , for different initial value there will be different result.
3) This algorithm is easy to be effected by abnormal points.
4) It may contain dead unit problem.

Table 1
List of research paper discussed

	Authors	Corresponding research paper
1	Ahamed Shafeeq B M and Hareesha K S	Dynamic clustering of data with modified K-Means algorithm
2	Ran Vijay Singh and M.P.S Bhatia	Data Clustering with Modified K-means Algorithm
3	Shi Na and Liu Xumin and Guan yong	An Improved k-means Clustering Algorithm

II. DYNAMIC CLUSTERING OF DATA WITH MODIFIED K-MEAN

This paper propose a new algorithm which can increase the cluster quality and fix the number of cluster. In standard k-mean algorithm we have to give the value of k i.e the number of cluster in advance . Practically it is very difficult to give the value of k in advance or fixing the value of k in advance will lead to a poor quality cluster. If the value of k is very small then there will be a chance of putting dissimilar objects into same group and if the value of k is large
then the more similar objects will be put into different groups. In this algorithm we have to give the value of k as input and we also have to mention either the value of k is fixed or not fixed. This algorithm work for two cases

1) When the value of k is fixed.
2) When the value of k is not fixed.

In the first case the algorithm work as standard k- mean algorithm. In the second case the user give the minimum value of k. The algorithm calculate the new cluster center by increasing the value of cluster counter by one in each iteration until it reaches the cluster quality threshold .
The dynamic algorithm is as follows:

Input

K : number of cluster (for dynamic clustering initialize the value of k by two)
Fixed number of cluster = yes or no (Boolean).
D : {d1, d2,…..dn} a data set containing n objects.

Output

A set of k clusters.

Method

1) Randomly choose k data item from D dataset as the initial cluster centers.
2) Repeat
3) Assign each data item di to the cluster to which object is most similar based on the mean value of the object in cluster;
4) Calculate the new mean value of the data items for each cluster and update the mean value;
5) Until no change.
6) If fixed number of cluster = yes (Go to step 12)
7) Compute the inter-cluster distance using eq (1)
8) Compute the intra-cluster distance

9) If new intra-cluster distance<old intra –cluster and new inter-cluster distance > old inter-cluster distance goto step 10 else goto step 11.
10) k=k+1
11) Stop

This algorithm give optimal number of cluster for unknown data set. The time taken by this algorithm for small dataset is almost same as standard k-mean algorithm but the time taken by dynamic clustering algorithm for large data set is more as compare to standard k-mean algorithm.

III. DATA CLUSTERING WITH MODIFIED K-MEANS ALGORITHM

This paper propose a new approach of data clustering based on the improvement of sensitivity of initial center of clusters. This algorithm will decrease the complexity and effort of numerical calculation and maintain the simplicity and easiness of implementing the k-mean algorithm. This algorithm can reduce the two limitation of standard k-mean algorithm . First is in standard k-mean result directly depends on initial centroid of cluster chosen by algorithm. Second limitation is it may contain dead unit problem. The proposed algorithm is based on density of different region which reduce the first problem. It will also solve the dead point problem because the center of cluster located in first iteration pertaining to the maximum density of data point. In this algorithm the number of desired cluster(k) is provided by user in same way as in standard k mean algorithm . They divide this approach in two phases:

Phase 1:

In first phase we take the data set and desired number of cluster(k) as input. The whole space is divided in to k*k segment. For example if value of k entered by user is 3(k=3) then the space will be partitioned in to 3*3 segment (3 segment horizontally and 3 segment vertically).Distribute the whole dataset in space and then calculate the frequency of data point in each segment. The segment which have the maximum probability of data point will have the maximum probability to contain center of cluster. If highest frequency of data point is same in two segment and upper bound of segment crosses the threshold then these two segment must be merged and then take the highest k segment for calculating the seed point of cluster. For example if value of k entered by user is 3(k=3) then the space will be partitioned in to 3*3 segment (3 segment horizontally and 3 segment vertically).

Phase 2:

Assign the data point to appropriate cluster center

Step1 : Calculate the distance between each cluster's center by using equation
$|C_i, C_j|=\{d(m_i, m_j) : (i, j) \epsilon [1,k] \& i \neq j\}$……(A)
Where $|C_i,C_j|$ is the distance between cluster C_i and C_j.
C_i : is the i^{th} cluster.
K : number of cluster centroid.
Step2 : Calculate the half of minimum distance between a centroid to the remaining centroid.
$dc(i) = ½(\min\{|Ci,Cj|\}$……..(B)
where
$d_{c(i)}$: is the half of minimum distance from i^{th} center to any other remaining cluster.
Step3 : select any data point to calculate its distance from i^{th} center and called this distance as d and compare it with $d_{c(i)}$.
If $(d<=d_{c(i)})$ then
Assign data point to i^{th} cluster.
else
calculate the distance from other centroid.
Repeat this process until that data point is assigned to remaining cluster.
If data point is not assigned to any cluster then the center which show minimum distance with the data point becomes the cluster for that data point. Repeat the process for each data point
Step4 : calculate the mean of each cluster and update it .
Repeat 2^{nd} phase until termination condition is reached.

MODIFIED ALGORITHM

1) Input data set and value of k
2) If(k=1) then exit
3) Else
4) /* divide data point space in k*k segments*/
5) For each dimension{
6) Calculate the minimum and maximum data points in each segment
7) Calculate range of group (R_G)=((min+max)/k)
8) Divide the data point space in k group with width R_G
9) }
10) Calculate the frequency of data point in each segment.
11) Choose k highest frequency group
12) Calculate mean of the selected group /* this will be initial center of cluster*/
13) Calculate distance between each cluster using eq. A
14) Take the minimum distance for each cluster and make it half using eq. B
15) For each data point P=1 to N_0\{
16) For each cluster j=1 to k {
17) Calculate $d(Zp, Mj) = \sqrt{\sum_{k=1}^{s}(Zp, k - Mj, k)}$
Where
Z_p is the p^{th} data point
M_j is the centroid of j^{th} cluster.
18) If $(d(Z_p, M_j) \leq dc_j)$\{
19) Then Z_p assign to cluster C_j
20) Break
21) }
22) Else
23) Continue;
24) }

25) If Z_p does not belong to any cluster then
26) $Z_p \epsilon$ min(d(Z_p, M_j)) where i ϵ [1, N_c]
27) }
28) Check termination condition of algorithm if satisfied
29) Exit
30) else
31) Calculate centroid of cluster
$M_j = 1/n_j(\sum Z_p)$
$\Delta Z_p \epsilon C_j$
Where n_j is the number of data point in cluster j
32) Goto step 13

In this algorithm step 5 to 12 is one time execution step it reduce the problem of dead unit and optimize the selection of initial centroid of cluster by using most populated area as center of cluster. As in modified algorithm threshold distance is used step from 13 to 27 will ensure minimum execution time during the allocation of data point to cluster.

IV. AN IMPROVED CLUSTERING ALGORITHM

This paper propose a new algorithm which increase the speed and accuracy of clustering and reduce the computational complexity of standard k-mean algorithm. As in k-mean algorithm in each iteration we have to calculate the distance between each data item and all cluster centers and then find the nearest cluster center and assign data item to that center. It reduce the efficiency of k-mean algorithm especially for large capacity data-bases. The basic idea of this algorithm is two keep two simple data structure to store the information of each iteration and that information can be used in next iteration. First data structure can be used to store the label of cluster. Second data structure can be used to store the distance of each data item to the nearest cluster center in each iteration this information can be used in next iteration. In second iteration we calculate the distance between data item and the new cluster center. After that we compare the distance between data item and new cluster with distance stored in previous iteration. If the new distance is smaller than or equal to older center then data item stay in its cluster that was assigned in previous iteration. Now there is no need to calculate distance between data item and remaining k-1cluster center. Some data item will remain in original cluster in each iteration so there is no need to calculate the distance and it will reduce the computational complexity.

ALGORITHM:

Input

K : number of desired cluster
D : {d1, d2,…..dn} a data set containing n objects.

Output

A set of k cluster as specified in input

Method

1) Arbitrarily choose k data item from D dataset as initial cluster centriod;
2) Calculate d(d_i, c_j) = the distance between each data item and all k cluster center. Assign data item to nearest cluster.
Where d_i : is data item in data set D
(1<=i<= n)
C_j : is the cluster center
(1<=j<= k)
3) For each data object d_i find the closest center c_j assign d_i to cluster j.
4) Store the label of cluster center in which data object d_i is in array cluster[]. Store distance of data object d_i to the nearest cluster in array dist[].
Set Cluster[i] = j, j is the label of nearest cluster.
Set Dist[i] = d(di, cj), d(di, cj) is the nearest Euclidean distance to the closest center.
5) For each cluster j(1<=j<=k) calculate cluster center;
6) Repeat
7) For each data object d_i compute its distance to the center of present nearest cluster
a. If this distance<=Dist[i] then
Data object stay in initial cluster;
b. else
for every cluster center c_j compute distance d(di, cj) of each data object to all the center assign data object d_i to nearest cluster center c_j.
set cluster[i]=j;
set Dist[i]=d(d_i, c_j);
8) For each cluster center j recalculate the centers;
9) Until convergence criteria met
10) Output the clustering results;

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.

V. CONCLUSION

This paper analyze the shortcomings of k-mean algorithm and also discuss three dissimilar algorithms that remove the limitations of k-mean algorithm and improve the speed and efficiency of k-mean algorithm and result in optimal number of cluster. The first algorithm remove the limitation of specifying the value of k in advance which is very difficult practically. It also gives the optimal number of cluster. The second algorithm reduce computational complexity and also remove dead unit problem. It select the most populated area as center of cluster. In k-mean algorithm result will depends on initial centroid second algorithm also reduce this problem. Third algorithm reduce the time complexity by using two simple data structure to store the information of each iteration which can be used in next iteration. As in first algorithm time complexity is greater as compared to standard k-mean algorithm for large data set .

Hence it can be concluded that if we use idea of third algorithm i.e we use data structure to store information in first algorithm we can reduce the time complexity of that algorithm and result in optimal solution.

REFERENCES

[1] Ahamed Shafeeq B M, Hareesha K S "Dynamic Clustering of Data with Modified K-Means Algorithm" International Conference on Information and Computer Networks vol. 27, pp.221-225, 2012 .
[2] Shi Na, Liu Xumin, Guan yong "Research on k-means Clustering Algorithm" Third International Symposium on Intelligent Information Technology and Security Informatics,IEEE 2010, pp. 63-67.
[3] Ran Vijay Singh, M.P.S Bhatia "Data Clustering with Modified K-means Algorithm" IEEE-International Conference on Recent Trends in Information Technology (ICRTIT), pp. 717-721, 2011.
[4] Jian Zhu, Hanshi Wang "An improved K-means Clustering Algorithm" 2010 IEEE
[5] Han J. and Kamber M. "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers, San Francisco, 2000.
[6] 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.
[7] Sun Shibao, Qin Keyun,"Research on Modified k-means Data Cluster Algorithm" Computer Engineering, vol.33, pp.200– 201,July 2007.
[8] D. Napoleon, P. Ganga lakshmi "An Efficient K-Means Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Point" IEEE 2010, pp. 42-45.