© 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
National Conference on Advanced Research Trends in Information and
ng Technologies(NCARTICT-2018), Department of IT,L. D. College of Engineering,
Ahmedabad, Gujarat, IndiaIn association with
International Journal of Scientific Research in Science, Engineering and Technology

An Improved K-Means Clustering Algorithm

Ekta Joshi[1], Dr. D. A. Parikh[2]
[1]Computer Engineering, L.D. College of Engineering, Ahmedabad, India
[2]HOD Computer Engineering, L.D. College of Engineering, Ahmedabad, India

ABSTRACT

This Vast spread of computing technologies has led to abundance of large data sets. Thus, there is a need to find similarities and define groupings among the elements of these big data sets. One of the ways to find these similarities is data clustering. Currently, there exist several data clustering algorithms which differ by their application area and efficiency. Increase in computational power and algorithmic improvements have reduced the time for clustering of big data sets. But it usually happens that big data sets can’t be processed whole due to hardware and computational restrictions. Clustering techniques, like K-Means are useful in analyzing data in a parallel fashion. K-Means largely depends upon a proper initialization to produce optimal results.

Keywords: K means, Clustering, Data Mining, Big Data.

I. INTRODUCTION

There has been a tremendous growth in the volume of data in the recent times. Data, whether it be structured or unstructured contribute to this enormous collection. To draw meaningful insights from this mountain of data we need algorithms which can perform analysis on this data. Clustering is the process of grouping data into groups called clusters, so that the objects in the same cluster are more similar to each other and more different from the objects in the other group [1]. It is one of these various important analysis techniques that is employed to large datasets and finds its application in the fields like search engines, recommendation systems, data mining, knowledge discovery, bioinformatics and documentation to name a few.

Nowadays, the data being generated is not only huge in volume, but is also stored across various machines all around the world. We need to process this data in parallel to reduce the cost of processing. K-Means is one of the most famous algorithms in the field of data mining [6]. Its scalability to large datasets and simplicity can be considered as one of the major reasons for its popularity. It is simple in data analysis and provides good performance. But it has a great dependence on the initial cluster center. The selection of initial cluster centers determines the quality of clustering. Therefore, it is an important step to select a reasonable set of initial cluster centers in K-means algorithm.

II. THE TRADITIONAL K - MEAN ALGORITHM [6]

K-means algorithm is a clustering algorithm based on partition, proposed by McQUeen in 1976. The aim of Kmeans algorithm is to divide M points in N dimensions into K clusters so that the precision rate and the recall rate are maximum. It is not practical to require that the solution has maximum against all partitions, except when M, N are small and K=2. The algorithm seeks instead of “local” optima solution, such that no movement of an object from one cluster to another will reduce the within-cluster sum of squares.

The basic principle of the traditional K-means algorithm is: firstly, each data object in the data set is regarded as a single cluster, randomly select K data objects as the initial clustering centers; secondly, successively calculate the distance of the rest data objects to each of the K cluster center, each data object will be categorized into the nearest cluster, and then recalculate the centroid of each cluster; repeat iteratively until the cluster partition is no longer changed. The process of K-means algorithm is as follows:

Input: data set contained n data objects, k(the number of clusters) ;
Output: k clusters;
Step1: Randomly select K data objects as the initial cluster centers;
Step2: Calculate the distances from the remaining data objects to initial cluster centers, assigned the remaining n-k data objects to the nearest cluster;
Step3: Recalculate the cluster centers of each cluster;
Step4: repeat step2 and step3 until convergence;

K-means algorithm is a simple and efficient clustering algorithm [6]. Its time complexity is close to O(n*k). When the differences between categories are small or the scale of data set is large, K-means algorithm will perform more efficient, and get better clustering results. It has two major drawbacks- (1) A priori fixation of the number of clusters (2) Random selection of initial centers. So there are different methods to improve the algorithm while maintaining its simplicity and efficiency.

III. TYPE OF METHODS

A. K-mean+:a developed clustering algorithm for big data [2]

In this paper, it proposed a new approach for fast clustering. It divides first instances into blocks applying block operation.

Block operation: let dataset D has M attributes and N instances for each attribute; range is divided by f equal width. The feature space of D is separated to blocks of a size. N instances are assigned to these blocks and processed as one instance but weighted by number of instances in single block.

Distance calculation: location of block is decided by the center of a block and then Manhattan Distance as function of distance measure instead of Euclidean distance, causes much floating point arithmetic.

Iteration: Cluster center and cluster labels of block update is iterated. When cluster center do not change, iteration stops. The result is rounded to integer to reduce complexity as cluster centers are weighted average of blocks.

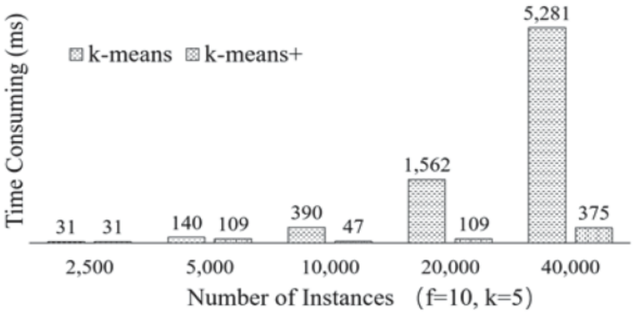

Figure 1. Time complexity [2]

B. Batch Clustering Algorithm for Big Data Sets [4]

In this paper, it is proposed to cluster a given large data set in batches by using k-means algorithm. That is take some portions of data elements from the given data set and process it. Then take next portion and process it and so on until all the elements of data set are processed. After that the whole given data set is also clustered by k-means algorithm for efficiency and quality comparison. Later qualitative indicators for both of these approaches are measured. For qualitative indicators the below are considered:

- Time T required for calculation of centroids and assigning data elements to centroids (clusters);
- The value of the objective function J (squared error function):

$$j = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2$$

Where $\left\|x_i^{(j)} - c_j\right\|^2$ is a given distance (Euclidean) measure between a data element $x_i^{(j)}$ and the cluster center c_j , i.e. j is an indicator of the Euclidean distance of the n data elements from their corresponding cluster centers.

Assuming that we have a data set consisting of n data elements, m is a number of elements in a small subset of data set, k is a number of centroids, the proposed batch clustering algorithm can be defined as follows:

1. We take m number of data elements from the given data set of n elements (m < n);
2. m number of data elements are processed in RAM of computer by the k-means algorithm to find k number of centroids (k < m);
3. Then take the k centroids along with m − k number of data elements from the remaining data set;

4. Repeat steps 2 – 3 until all the elements of the initial data set are processed;
5. Assign all elements to the k centroids calculated in the last step 4.

The result of calculations is below:

Classic k-means
- Time spent calculating centroids: T = 225 sec
- Time spent calculating objective function: T = 979 sec
- Objective function: J = 33997

Batch clustering
- Number of elements in each portion: m =100,000 elements.
- Time spent calculating centroids: T = 223 sec
- Time spent calculating objective function: T = 932 sec
- Objective function: J = 33996

As seen from the above calculation results, batch clustering algorithm produces better results over classic k-means algorithm. Besides, we have a big gain in using computational power with restricted resources.

C. New Approach for Clustering of Big data: DisK-means[1]

The proposed new algorithm called DisK-Means. The traditional K-Means algorithm produces varying results over several runs on large datasets. It is also very time consuming. Our algorithm reduces the time of execution along with improving the quality of clusters.

Our algorithm divides the complete dataset into *m* parts, where each part must have more than minimum the sample size to be representative of *Y*. It is not necessary that it should be equal to the computational cores available as in CK-Means. In step 2, for each part obtained from step 1, a distance matrix [15] is made which contains the shortest distance from each point to every other point in the set. *D(i,j)* represents the shortest distance from point *i* to *j* and *M(i,j)* represents the matrix that has *i* rows and *j* columns. Now for each subset, choose the point that has the minimum sum of distances. In step 5, K-Means++ is performed on each subset obtained in the initial stage. Here, the initial points for each subset are not chosen randomly but from each subset in the previous step. Next, fitness measure of each cluster is calculated and stored in the array *Ti*. We use WSSQ method as it is easy to calculate and does not add further complexity. The cluster that has the minimum value of WSSQ is the best fit and its centers are put in the set *C* which contains the initial centers for the next step. The re-clustering is further performed using K-Means to finally obtain *k* centers.

Algorithm 3 DisK-Means [1]

```
1. Partition y into y1, y2,..., ym ;
2. For each i ∈{1,2,.....m} do
3. Calculate the distance D(i,j), from each point to every
   other point and form the distance matrix M(i,j).
4. Find the node with minimum sum of distances and put
   it into ICi;
5. Run K-Means++ on yi using ICi as the initial node to
   get k centroids Ci' and clusters (clyi);
6. Ti ←f(clyi);
7. C=Ci' where i←Best-fit;
8. Re-clustering using K-Means with C as set of initial
   centers.
```

TABLE 1. COMPARISION BASED ON RUNNING TIME [1]

S.no.	Running Time (in sec)		
	Values of k	K-Means	*Disk-Means*
1	50	300	220
2	50	340	223
3	100	200	127
4	100	213	130

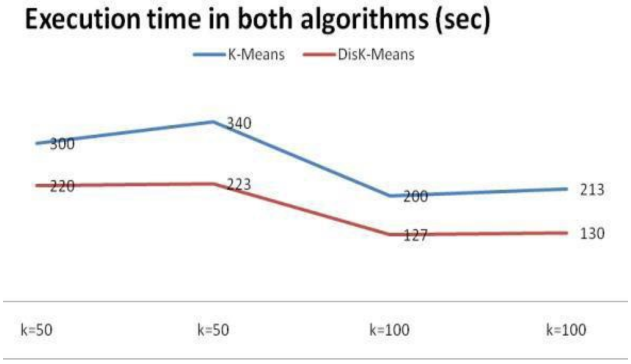

Figure 2. Execution time for K-Means and DisK-Means [1]

It can be observed that in both cases, when k=50 and k=100, our new DisK-Means takes less time to execute with less deviation when compared to K-Means. A major drop is observed when the value of k is changed dramatically from 50 to 100. It is expected that as the number of clusters increase, the time for execution would decrease.

D. A hybrid clustering algorithm : the FastDBSCAN [7]

Proposed FastDBSCAN including two steps:

(1) To Partition the data set by K-means and then use Min-Max method to sample data. The main principle of min-Max is: first a starting point y1 is randomly chosen from the dataset D. Then all the other points in Y are chosen among the points of dataset X that maximize their minimal distance from the point already in Y. Thus when t points already belong to y, the process that selects the point y_{t+1} from X can be formalized as shown in equation below:

where d(.) denotes the distance defined in the space of the objects. In each iteration methods selects the point that exhibits the largest label uncertainty according to the previous answers of the user.

```
Algorithm 1: Min-Max method;
Input: data set D, the number of samples k;
Output: set of points selected by Min-Max Y.
1. Take any reference point r;
2. Insert r in Y
3. Temp = 1;
4. while |temp|≤ k+1
5. Find the point x that maximize their minimal distance from the points already in Y
6. Insert x in Y
7. temp = temp +1
8. endwhile
9. remove r from Y
10.return Y
```

(2) Clustering sampled data by FastDBSCAN. K-means in first step guarantee that the data chosen for step 2 will cover the whole data set. Then it extracts t percent of points by Min-Max method. This new set is used by DBSCAN.

```
Algorithm 2: FastDBSCAN
Input: A data set D, the number of clusters for K-Means k, the proportion of data t;
Output: Clusters and noises.
1. Initialize k centers
2. Partition data by K-Means,
3. Take a proportion t of points (Min-max algorithm) from clusters to form a new data set E; build a correspondence list to associate each selected point with its cluster.
4. Perform DBSCAN clustering on the set E,
5. Recover the clusters detected by DBSCAN to form final clusters.
```

Comparison of FastDBSCAN and DBSCAN is with two aspects: clustering accuracy and time calculation. The accuracy of the algorithm is better as k-means is used for clustering in the step one. The time calculation of FastDBSCAN is less compared to DBSCAN.

E. An Improved K-means text clustering algorithm By Optimizing initial cluster centres[5]

The basic idea of the improved K-means algorithm is: at first, calculate the density parameter of all data objects in the data set, and determine data objects which are isolated points.

If a data object is isolated, it will be removed from the data collection. After deleting the isolated points, we will get a data set with high density parameter. Then, select a data object with the highest density parameter in the set as the first initial cluster centre; Next, select a data object from the rest of high-density data collection as the second initial cluster centre, which is the furthest from the first initial cluster centre; And so on, until find k initial cluster centres. Based on this k initial cluster centres, use the traditional K-means algorithm to do clustering.

The process of the algorithm is described as follows:

Input: text set D = {d1, d2, ··· , dn} containing n data objects, and k (the number of clusters);

Output: k clusters;

Step1: Calculate distances between any two data objects in data set D and the average distance, using formula (1) and (2) respectively;

Step2: Calculate density parameters of all data objects in the data set D and the average density parameter of the set D, using formula (3) and (4);

Step3: According to the formula (5), determine isolated data objects, and delete them from the set D, thus obtain a collection A with high density parameter;

Step4: Select a data object with the highest density parameter from collection A as the first initial clustering center, and add it to the collection B, and remove it from collection A;

Step5: From collection A, select a data object which is furthest from collection B as the next initial cluster center, and add it to collection B, and remove it from collection A;

Step6: Repeat Step5, until the number of data objects in collection B is k;

Step7: Based on the k cluster center, use the traditional Kmeans to do clustering;

This algorithm is divided into two stages, one is to determine the initial cluster centers, and the other is to use the traditional K-means to do clustering based on the initial cluster centers. To determine the initial cluster centers needs to calculate the distances between all data pairs, and its time complexity is O(n2). Because of the extra computation, this algorithm's time complexity is higher than that of the original algorithm, but it can get a better clustering result.

TABLE 2. Precision Evaluation [5]

Clustering algorithm	**Evaluation index**	**Art**	***Economy***	***Environment***	***Political***	***Sports***
K-MEANS	P (%)	80.33	78.84	79.53	80.95	78.62
	R (%)	81.14	82.45	78.97	81.83	79.69
Improved k-mean	P (%)	81.74	80.42	81.21	82.71	81.58
	R (%)	82.23	83.59	82.36	83.75	82.64

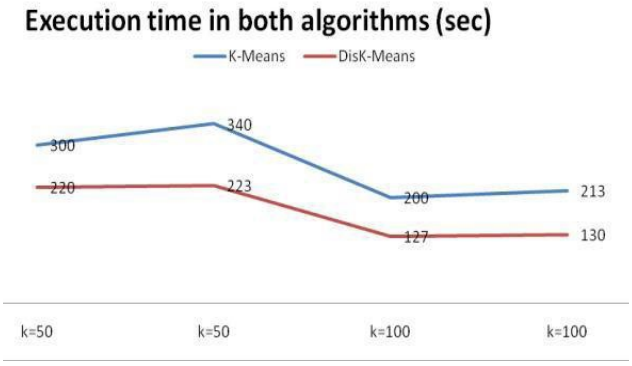

Figure 3. Execution time for K-Means and DisK-Means [5]

IV. ANALYSIS

In this paper, a study of different types of algorithms of k-means modified been represented each have improved the one or the other shortcomings of the k-means algorithm and hence provided a better and efficient algorithm to learn a large data set. A verity of methods are been introduced want to tackle a large amount of dataset and provide a fast clustering while keeping the spark of k-mean simplicity and efficiency.

V. CONCLUSION

The problem of k-mean clustering algorithm has a lot of shortcomings: (1)It requires the user to specify the number of clusters in advance. However, in the beginning, the user does not know how many clusters should be divided into. (2) It has a great dependence on the initial cluster center, and it is easy to produce the local optimal solution. Since K-means' clustering criterion function is a non-convex squared error evaluation function, which leads to there is only one global minimum, but there are a number of local minimum. The randomly selected initial clustering centers tend to fall into the non-convexity, causing the algorithm deviates from the searching range of global optimal solution. So, when the initial clustering centers are selected improperly, the clustering results will be unstable and inaccurate. (3)It is sensitive to isolated points and noise data. K-means algorithm takes average point as cluster's center, and adds it to the next round of the algorithm, resulting in the cluster's center may be away from the dense regions of data set, and the cluster's center may be a noise point or an isolated point. Therefore, if the data set contains a lot of isolated points or noise data, to a great extent, the clustering results will be controlled by noise or isolated data, leading to the inaccurate or even wrong clustering results.

Different aspects of k-mean algorithm are deal with and different approaches are introduced working on different issues of k-means clustering algorithm. One has reduced time to response, other to reduce the complexity of algorithm using k-mean. One of papers is dealing with a large amount of data dividing the work into batches. All these methods enhanced traditional k-mean algorithm to an efficient algorithm.

VI. REFERENCES

[1] Anu Saini, G. B. Pant ,Jaypriya Ubriani “New Approach for Clustering of Big Data: DisK-Means”, 2016 IEEE ,International Conference on Computing, Communication and Automation ,pp 122-126;

[2] Kun niu, zhipeng gao,haizhen jaog ,haijie deng “K-mean+:a developed clustering algorithm for big data”, 2016 IEEE , Proceedings of CCIS2016,pp 141-144;

[3] Vadlana Baby,Dr. N. Subhash Chandra “Distributed threshold k-means clustering for privacy preserving data mining”,2016 IEEE,Conference on Advances in Computing, Communications and Informatics (ICACCI);

[4] Rasim Alguliyev , Ramiz Aliguliyev , Adil Bagirov , Rafael Karimov “Batch Clustering Algorithm for Big Data Sets”;

[5] Caiquan Xiong, Zhen Hua, Ke Lv, Wuhan Hubei ,“An Improved K-means text clustering algorithm By Optimizing initial cluster centers”, 2016 IEEE, International Conference on Cloud Computing and Big Data,pp 265-268;

[6] Jiawei Han, Jian Pei, Micheline Kamber “Data Mining: Concepts and Techniques” 3rd edition;

[7] Vu Viet Thang, D.V. Pantiukhin, A.I. Galushkin “A hybrid clustering algorithm : the FastDBSCAN” 2015 International Conference on Engineering and Telecommunication,pp 69-74;

[8] Tahereh Kamali, Daniel Stashuk “A Density-Based Clustering Approach to Motor Unit Potential Characterizations to Support Diagnosis of Neuromuscular Disorders” 2016 IEEE Transactions on Neural Systems and Rehabilitation Engineering ;

[9] Bin Jiang, Jian Pei, Yufei Tao and Xuemin Lin, Member, IEEE “Clustering Uncertain Data Based on Probability Distribution Similarity” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 4, APRIL 2013;

[10] Chang Lu , Yueting Shi, Yueyang Chen, Shiqi Bao, Lixing Tang “Data Mining Applied to Oil Well Using K-means and DBSCAN” 2016 7th International Conference on Cloud Computing and Big Data;

[11] Jianbing Shen, Xiaopeng Hao, Zhiyuan Liang, Yu Liu, Wenguan Wang,and Ling Shao, Member, IEEE “Real-time Superpixel Segmentation by DBSCAN Clustering Algorithm” 2016 IEEE TRANSACTIONS ON IMAGE PROCESSING;

[12] Dongming Tang.Affinity propagation clustering for bid data based on Hadoop. Computer Engineering and Applications, 2015, 51(4):29-34;

[13] Joshua M.Dudik a, AtsukoKurosu b, JamesL.Coyle b, ErvinSejdić a,n “A comparative analysis of DBSCAN, K-means, and quadratic variation algorithms for automatic identification of swallows from swallowing accelerometry signals”, Computers in Biology and Medicine 59 (2015);

[14] Jesal Shethna “Data Mining Techniques available from https://www.educba.com/7-data-mining-techniques-for-best-results/” November 7, 2016;

[15] Martin Brown “Key techniques from https://www.ibm.com/developerworks/library/ba-data-mining-techniques/” Published on December 11, 2012;

[16] Data Mining tutorials “Data Mining Techniques from http://www.zentut.com/data-mining/data-mining-techniques/”

[17] Saurabh Arora, Inderveer Chana “A Survey of Clustering Techniques for Big Data Analysis” 2014 5th International Conference- Confluence The Next Generation Information Technology Summit (Confluence),pp 59-65.

[18] Martin Ester, Hans Peter Kriegel, Jorg Sander, Xiaowei Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, Published in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)