ISSN:2231-0711

A Modified K-Means Algorithm for Big Data Clustering

SK Ahammad Fahad[1*], Md. Mahbub Alam[2#]
[1]IBAIS University, Dhaka, Bangladesh*
[2#]DUET, Dhaka, Bangladesh

Abstract—**Amount of data is getting bigger in every moment and this data comes from everywhere; social media, sensors, search engines, GPS signals, transaction records, satellites, financial markets, ecommerce sites etc. This large volume of data may be semi-structured, unstructured or even structured. So it is important to derive meaningful information from this huge data set. Clustering is the process to categorize data such that data are grouped in the same cluster when they are similar according to specific metrics. In this paper, we are working on k-mean clustering technique to cluster big data. Several methods have been proposed for improving the performance of the k-means clustering algorithm. We propose a method for making the algorithm less time consuming, more effective and efficient for better clustering with reduced complexity. According to our observation, quality of the resulting clusters heavily depends on the selection of initial centroid and changes in data clusters in the subsequence iterations. As we know, after a certain number of iterations, a small part of the data points change their clusters. Therefore, our proposed method first finds the initial centroid and puts an interval between those data elements which will not change their cluster and those which may change their cluster in the subsequence iterations. So that it will reduce the workload significantly in case of very large data sets. We evaluate our method with different sets of data and compare with others methods as well.**

Keywords—**Big Data, K-means, Clustering, Clustering Algorithm, Modified K-means.**

I. Introduction

This Big Data has become one of the buzzwords in IT during the last couple of years. In the current digital era, according to massive progress and development of the internet and online world technologies such as big and powerful data servers, we face a huge volume of information. Data size has increased dramatically with the advent of today's technology in many sectors such as manufacturing, business, science and web applications. With the rising of data sharing websites, such as Facebook, Flickr, YouTube, Twitter, Google, Google news, there is a dramatic growth in the number of data. For example, Facebook reports 2.5 billion content items, 105 terabytes of data each half hour, 300M photos and 4M videos posted per day [16]. In Twitter, Over 651 million users, generating over 6,000 tweets per second [15]. 300 hours of video are uploaded to YouTube every minute with more than 1 trillion video views [14]. Google make 50 billion pages indexed and more than 2.4 million queries in every minute [13]. Google news Articles from over 10,000 sources in real time [16]. More than 4.5 million photos uploaded in a day in Flickr[16]. It is estimated that all the global data generated from the beginning of time until 2003 represented about 5 Exabyte and Over 1.8 Zettabytes(zb) created in 2011. Approximately 8 Zettabytes(zb) by 2015 [16].

These massive volumes of data can be useful to people for their business if we can assure the meaning of those data. Proper information can lead us to right path. The process to gather information from huge unstructured or semi-structured data is proper clustering. Proper clustering return clusters which contains similar characteristic elements. Thousands of clustering algorithms have been published since then; K-means is the widely used with a wide range of applications. Different modifications have been proposed in the literature for improving the performance of k-means clustering algorithm. Our Proposed method first calculates the initial cluster. Then we get initial centroid from this initial cluster. After that native K-means algorithm works based on initial cluster and centroids. But proposed algorithm does not calculate all data points like native K-means because we use intervals to select the point’s that have the chance to change their cluster i.e. this method calculate only the inner interval points.

The rest of the paper organized as follows: Section II illustrates the related works. Our proposed algorithm with details modification fields explain in section III. In section IV, we discuss the experimental results and comparison with other methods. Finally, we conclude the paper in section V.

II. Related Works

To get more 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. Krishna and Murty[4] 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[1] 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. Forgy’s method [2] 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. Likas et al. [5] 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.

Bradley and Fayyad [3] presented a refined algorithm that utilizes K-means M times to M random subsets sampled from the original data. The most common initialization was proposed by Pena, Lozano et al. [6]. This method is selecting randomly K points as centroids from the data set. The main advantage of the method is simplicity and an opportunity to cover rather well the solution space by multiple initialization of the algorithm. Ball and Hall proposed the ISODATA algorithm [7], which is estimating K dynamically. For selection of a proper K, a sequence of clustering structures can be obtained by running K-means several times from the possible minimum Kmin to the maximum Kmax[12]. These structures are then evaluated based on constructed indices and the expected clustering solution is determined by choosing the one with the best index [8]. The popular approach for evaluating the number of clusters in K-means is the Cubic Clustering Criterion [9] used in SAS Enterprise Miner.

This paper proposed a modified k-means clustering algorithm based on initial cluster and centroids with an effective interval. Initial centroids and initial cluster data used as input to stander k-means algorithm and then effectively use the points that have the chance to change the cluster which will provide better performance than other methods.

III. Proposed Methodology

There have been a lot of research works done in the past to improve the K-Mean clustering technique. All they wanted to improve the clustering result and fix the limitations of previously proposed method. We get influenced by those thesis and research works and decided to make some modification that can more efficient and faster.

We worked on the problem to find initial cluster(R). Also we worked to find initial centroid. In last part of our algorithm, we try to minimize calculation by finding the feasible points of working. When we combined all of these points, we found an effective modified k-means algorithm.

A. *Modified Algorithm*

Modified K-means algorithm:

Input: $U = \{n_1, n_2, n_3, \ldots\ldots\ldots\ldots\ldots. n\}$ // Set of n number of data points.

Output: A set of R Clusters. // Number of desired Cluster.

Steps:

I) Calculate $R \cong \sqrt{\frac{n}{2}}$.

II) Allocate $x = 1$, assign a value to f // if the value of f is not given than it was 1 by default and f > 0.

III) Find the closest pair of data point from$U$. Move those points to new set N_x.

IV) Find the closest point of N_x and move it to $N_x$from U.

V) Repeat step IV) until the number of element of N_x reach $(\frac{n}{R} \times f)$.

VI) When, N_x are full and the number of elements of $U \neq 0$. Increment the value of x. New $x = x + 1$. Repeat step III).

VII) The center of gravityof each N_x . Those are the initial centroids C_j and elements of N_x are the elements of R_m.

VIII) Find the closest centroid for each data points and allocate each data points Cluster. Calculate new centroid for each R_m.

IX) Calculate the D_l (D_l= largest distanced data point from each centroid of each cluster).

X) Get the data points in the interval of $D_l \times \frac{4}{9}$ to D_l .

XI) Find the closest centroid for those points and allocate them to closest centroid cluster.

XII) Find new centroid for eachR_m.

If, any change in any R_m . Repeat step IX. Otherwise go to end.

In the first part, we found the number of cluster. Then assign data points to initial cluster. Then we go for find an effective and useful initial centroid. In these two parts; there have some loops and its calculation but these make a better clustering rather than other clustering methods. For make algorithm more intelligent, some procedure must add. Intelligence of algorithm will help to determine the number of cluster and which points are the initial centroids. In last portion of modified algorithm, we worked only those feasible data points which have chance to change current cluster and move to new cluster.

We also make a short list of points for calculate. It minimizes calculation, that's why it was capable to save time on behalf of original standard k-means algorithm.

In step I, we calculate the equation and get a concept about the number of cluster. We take a concept about the cluster number, but this calculation is not the final decision. From step II, we assign a value in x for maintain cluster number. By step III, IV and V, algorithm assigns a new cluster and assigns data points to this cluster.

From step VI, algorithm finalizes the initial cluster's member data points, and gets decision to start a new cluster. Step VII; find the initial centroids for clusters. By help of step VII, step VIII finalized a stable cluster and centroid.

In step IX, X, XI and XII, there have tricks to find feasible data points those have chance to change current cluster. We worked only those interval's points. Normally other points don't move clusters. For this step, algorithm saves a lot of time. It minimizes a lot of calculation. To get decision, step XIII is used. In this step; algorithm take decision about the algorithm continue or all clustering is finished.

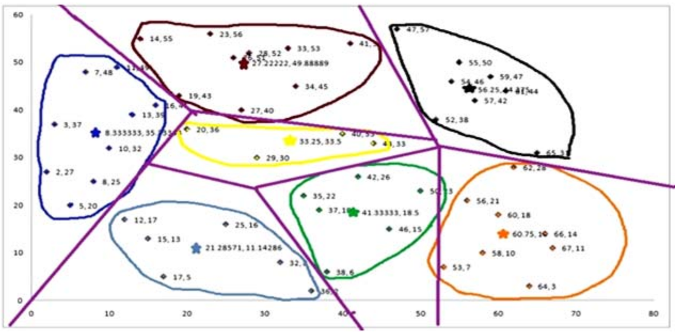

Figure 1. Result of 50 products

We apply on 50 products. Those have two characteristic, those are weight and size. Those are;

{(23,56),(11,49),(37,19),(28,52),(65,31),(44,33),(7,48),(13,39),(32,8),(40,35),(47,57),(60,18),(55,50),(17,5),(46,15),(34,45),(58,10),(50,23),(8,25),(26,51),(52,38),(19,43),(59,47),(56,21),(3,37),(29,30),(62,28),(67,11),(15,13),(12,17),(5,20),(14,55),(42,26),(66,14),(64,3),(53,7),(25,16),(36,2),(38,6),(10,32),(61,44),(41,54),(57,42),(35,22),(27,40),(2,27),(16,41),(20,36),(54,46),(33,53)}.

When we apply our algorithm on those 50 products, than we got result and, this result is given in Figure 1.

B. Field of Modification

K-means algorithms main limitation fields are; fix the value of K, Make an initial centroid. Lot of researcher worked these fields particularly. Our point of view is modified k-means as; it can overcome all limitations. Our concern of modification is combined all limitation and solved them for maximum output.

We solve the number of cluster problem. Algorithm can determine own its number of cluster by working data points. Our expression of determine number of cluster is $\cong \sqrt{\frac{n}{2}}$. Here, R= number of cluster. n= number of data inputted. It's the initial concept of Number of cluster. We fix the number of clusters in next equation. Basis of input f will be determined. If we apply k-means on human 'f' value is not equal to when we apply k-means on music instrument. If no value of 'f' is given than value of f is by default 1. Smaller than 1 value of 'f', increase the number of clusters. If the value of 'f' is bigger than 1, than the number of cluster is reduce.

When we get the number of cluster, our next target to get proper initial centroids. After get value of 'f', we compute the closest pair and keep them to N_x, and delete them to input set U. Than we fill N_x by calculate the nearest of those pair when any data goes to any N_x than it was deleted from U. When $N_x$is full we increase the value of x and continue same process as loop until in U have any data point left. Closest pair points mean value is initial centroids. Assign closest points of initial points into R_m. Those are initial cluster. Calculate the center of gravity to find the centroids. When we get each R_m center of gravity. Those are the centroids of each R_m. Now we have initial clusters and centroids.

When clusters and centroids are on our hand, we continue a loop for finalized cluster. In traditional k-means algorithm, all points are calculated, to get finalized cluster and their member. In this case, centroids move a little bit and closest points of center are unchanged. We keep this matter on think and we provide an interval. Calculate largest distanced data point D_l. $D_l \times \frac{4}{9} to D_l$is the interval. Inner interval points go for calculation. In each loop, calculate new centroids for Inner interval points. Assign inner interval points on their cluster and calculate new centroids. If any change happened in any cluster, do same process again and again. When no change in any cluster. We found our desire clusters. Those are the final cluster we want.

In first two portions we make our algorithm capable to find the number of cluster and initial centroids with initial cluster. In last part of algorithm we represent an interval. Only inner interval points go for calculate and re-allocate clusters. We keep a lot of points out side of interval. Those points cluster are unchanged. It reduces calculation and makes our algorithm faster than traditional k-means clustering algorithm.

IV. EXPARIMENTS

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 [11], Optimized Version of the K-Means Clustering Algorithm by Poteras , Miha˘escu and Mocanu [10]. We use C++ language to implement our algorithm. Experiments were conducted on a machine consisting of operating system windows8.1 on Intel i7-4700MQ CPU, 8 GB RAM. The following table illustrates the comparison of our methods with other methods.

TABLE I. TRAIN ALGORITHMS FOR TESTING

Algorithm	Data points	Number of cluster	Initial Cluster
Standard K-means Algorithm	Input by user	Input by user	Input by user or random
Improve k-means Algorithm	Input by user	Input by user	Calculate by algorithm
Optimized K-Means Algorithm	Input by user	Input by user	Input by user or random
Our Algorithm	Input by user	Calculate by algorithm	Calculate by algorithm

TABLE II. RESULT FOR DIFFERENT DATA POINTS

Algorithm	5k points(sec)	10k points(sec)	50k points(sec)	100k points(sec)
Standard	50.38	157.44	656.57	1137.42
Improve	39.29	111.78	485.53	818.94
Optimized	35.77	100.76	446.68	737.05
Our Algorithm	35.36	92.88	405.80	667.67

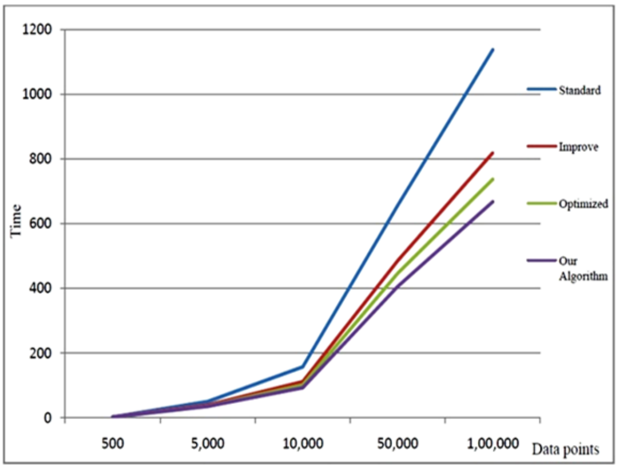

Figure 2. Process time of different algorithm for random dataset.

Four algorithms is train simultaneously by same data sets. Standard K-means Algorithm, Improve k-means Clustering Algorithm by Nazeer and Sebastian [11], Optimized Version of the K-Means Clustering Algorithm by Poteras, Miha˘escu and Mocanu[10] are need the value of k. Our algorithm can obtain value of k itself. Standard K-means Algorithm and Optimized Version of the K-Means Clustering Algorithm need the initial centroids information. Our algorithm and improve k-means clustering algorithm by Nazeer and Sebastian is capable to find initial centroids.

We use several dataset to testing the algorithm. We use 500 to 1, 00,000 random data points for testing.

After analyze the experimental result, we can decide our algorithm performance on large scale data clustering is much more batter and more effective rather than modern modification of k-means cluster algorithms.

REFERENCES

[1] 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

[2] Forgy E (1965) *Cluster analysis of multivariate data; efficiency vs. interpretability of classifications.* Biometrics, 21: pp 768-780

[3] Bradley P, Fayyad U (1998) *Refining initial points for K-means clustering.* International conference on machine learning (ICML-98), pp 91-99

[4] Krishna K, Murty M (1999) *Generic K-Means algorithm.* IEEE Transactions on systems, man, and cybernetics- part B: Cybernetics, 29(3): pp 433-439

[5] Likas A, Vlassis N, Verbeek J (2003) *The global K-means clustering algorithm*. Pattern recognition, 36(2), pp 451-461

[6] 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

[7] Ball G, Hall D (1967) *A clustering technique for summarizing multivariate data.* Behavioral science, 12: pp 153-155

[8] Milligan G, Cooper M (1985) *An examination of procedures for determining the number of clusters in a data set.* Psychometrika, 50: pp 150-179

[9] SAS Institute Inc., *SAS technical report A-108 (1983) Cubic clustering criterion.* Cary, NC: SAS Institute Inc., 56 pp

[10] CosminMarianPoteraş ¸MarianCristianMihăescu ¸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.

[11] 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.

[12] Rui X, Wunsch DC II (2009*) Clustering*. IEEE Press series on computational intelligence, John Wiley & Sons

[13] http://www.internetlivestats.com/google-search-statistics/

[14] http://www.internetlivestats.com/twitter-statistics/

[15] https://www.youtube.com/yt/press/statistics.html

[16] Anil K. Jain with RadhaChitta and Rong Jin, *Clustering Big Data*: Department of Computer Science Michigan State University.