Implementation of Enhanced K-means algorithm using OPENMP

[1]Prateek S Swamy, [2]M M Raghuwanshi, [3]A A Gholghate
[1]Student, [2]Professor, [3]Professor
[1]Computer Science and Engineering,
[1]Rajiv Gandhi College of Engineering and Research, Nagpur, India

***Abstract*— Serial execution of K-means algorithm on large dataset takes more execution time and does not give accurate results. Parallel processing is one of the ways to improve the performance of K-Means algorithm. But the execution time and accuracy is largely dependent on selection of initial cluster centers. In this paper, parallel processing of K-Means is proposed using an initialization method to originate initial cluster centers. Results have shown that time taken for clustering using our proposed approach is optimized as compared to serial K-Mean approach.**

***Index Terms*— Serial execution, large dataset, Parallel processing, K-Means, execution time, initial cluster centers.**

I. INTRODUCTION

Clustering of data is a key technique for analysis of data, which is used to find the similarity or dissimilarity between groups of item in a dataset such that item in one group are more similar than other group and vice versa[1].Because of modern methods for scientific data collection, size of database is increasing day by day. As a result, data mining is getting practically difficult by using conventional techniques [2]. An efficient algorithm for mining of data is the need of the hour so that useful information from large databases can be extracted.

Large number of algorithms has been developed for data clustering task. K-Means is the oldest and probably the most popular algorithm proposed for data clustering task [3].It is easy, efficient, fast and sensitive. But, K-Means algorithm has some issues [2, 4, 5, 6, 8]. These are

- Initialization of initial cluster center is random, which affects the accuracy.
- Serial execution takes more time
- No information about number of clusters in the dataset.
- Execution time depends on the number of cluster.

In this paper, parallel processing of K-Means is proposed using an initialization method to originate initial cluster centers, which not only reduces the execution time but also gives accurate results. Hence, a novel method is proposed which tries to improve K-Means by using an initialization method to address the first issue [7] and parallel processing using OpenMP [8] to address the second issue.

II. BACKGROUND REVIEW

A. K-MEANS CLUSTERING ALGORITHM

Pseudocode for the k-means algorithm [9] is given as follows:

Input: D= {d1, d2….. dn} //set of n data items
K //Number of desired clusters
Output: A set of K clusters
Steps:
1. Arbitrarily choose K data items from D as initial centroids
2. **Repeat**
2.1 Assign each data item di to the cluster which has the closest centroids.
2.2 Calculate the new mean of each cluster,
Until convergence criterion is met.

It can be observed from the above algorithm that initial centroids are chosen randomly. This affects the accuracy of algorithm [2]. In case if the dataset is large, serial execution of K-means takes more time [8]. These are the two important drawbacks of k-means clustering algorithm. To improve the accuracy of k-means, the initial centroids are originated by using binary search technique [7]. To speed up the execution time of k-means, parallel processing can be used Using OpenMP [8]. A given task is broken down into discrete parts and parallel execution is done with the help of child process. Execution of child process is simultaneous on different CPUs [10, 11].

B. OpenMP (Open Multi-Processing)

The OpenMP Application Programming Interface is one of the best emerging standards for parallel programming on shared-memory multiprocessors. It extends existing languages such as FORTRAN and C/C++ with a set of directives. To use parallelism with the code in OpenMP, the compiler directives are used. [12].

III. Related Work

A lot of efforts have been made by researchers to improve the accuracy and efficiency of k-means algorithm. K.A Abdul Nazeer et al. [2] have proposed a heuristic technique to find better initial centroids but the time complexity of the algorithm is not enhanced in this method. Moreover, if the number of attribute is more, then there is a chance that efficiency of the algorithm can be affected.

Yuan et al. [13] have proposed a technique to find out the initial centroids. The centroids obtained by this method produce clusters, which are more accurate than that of original k-means algorithm. But, efficiency of k-means is not improved using Yuan's method. Fahim A M et al. [14] proposed a better approach for assigning data-points to clusters. The original k-means algorithm is computationally very expensive because there is a need to calculate the Euclidean distance between data points and all preliminary centroids. In Fahim's approach, two distance function are used, one similar to k-means algorithm and another one based on prediction is used to minimize the number of distance computations. But, in this method, centroids are selected randomly, which affects the accuracy of final clusters.

DS Bhopal Naik et al. [8] have proposed a technique in which parallel processing of Enhanced K-means using OpenMP is done. Using this approach, the time taken for parallel processing of Enhanced K-means is optimized as compared to the serial approach. But, approach used in the selection of initial centroids is a heuristic one and hence it is not that accurate. Yugal Kumar et al. [7] has proposed a new initialization method to originate initial cluster centers for k-means algorithm based on binary search technique. The accuracy obtained in this method is impressive as compared to other method. But the time taken is more because the method is implemented in sequential manner.

IV. Proposed Approach

In this section, parallel processing of k-means algorithm using OpenMP is introduced [8] with an efficient method to find initial centroid [7].Results have shown that combination of these two approaches is new technique in the field of clustering of data. Using this approach, clustering time can reduced to a great extent.

Number of cluster **K** and dataset is given as an input. Then, initialization of OpenMP process is done to perform parallel processing in multi-core systems. Now, master process calculates value of initial centroids and broadcast it to the slave process. It has been assumed that number of process is equal to number of cores. We are using a dual core system in our implementation. In each slave process, data objects are assigned to nearest cluster and new mean of each cluster is calculated .This activity continues until convergence is met. Master process gathers result obtained at each **slave** process and finally master process declares the final **K** clusters

Values of initial centroids are calculated using an initialization method, which is based on the unique property of binary search algorithm [7, 16]. In binary search algorithm, the value of middle item in list is calculated as follows:

$$A[mid] = A[beg] + A[end] /2. \quad (1)$$

The above property is modified to find initial cluster points for K-means algorithm.

- A [beg] is replaced by A [max]
- A [end] is replaced by A [min]
- 2 is replaced by K , numbers of clusters
- A [mid] is replaced by any variable such as M
- Plus symbol is replaced by minus symbol

Now, the equation (1) is formulated in another equation as given below:

$$M = A[max] - A[min] /2. \quad (2).$$

The generalization form of the equation (2) can be written as:

$$M_i = max (A_i) - min (A_i)/K \quad (3).$$

The equation (3) is used to calculate the value of the variable M that specifies the range of initial cluster centers but not give the cluster centers

The cluster centers for K-Means algorithm are generated using given equation.

$$Ck = \min(Ai) + (K-1) * M \quad (4).$$

Consider an example dataset D that is given in Table I. The given dataset is applied with proposed method to get the initial cluster points. This dataset is consist total number of instances (N) = {9}, no. of attributes (i) = {2} and number of Clusters (K) = {3}. The working of proposed method is given below:

Objects	X1	X2	X3	X4	X5	X6	X7	X8	X9
A	1.1	1.3	1.2	3.2	2.8	2.9	2	1.9	2.2
B	4.3	3.9	3.8	4.8	3.9	3.7	3.6	3.3	3.2

Table I: Example dataset to generate the initial cluster center

•Calculate the maximum and the minimum values of each attribute in the dataset.

Maximum = (3.2, 4.8) and Minimum = (1.1, 3.2)

• Calculate the value of M as

M= {(3.2 – 1.1)/3, (4.8 – 3.2)/3}

M= {0.70, 0.53}

• Generate the initial cluster centers for initialization as

C1 = (1.1 + ((1-1) * 0.70), 3.2 + ((1-1) * 0.53)) = (1.1, 3.2)

C2 = (1.1 + ((2-1) * 0.70), 3.2 + ((2-1) * 0.53)) = (1.8, 3.73)

C3 = (1.1 + ((3-1) * 0.70), 3.2 + ((3-1) * 0.53)) = (2.5, 4.26)

The newly generated cluster centers (1.1, 3.2), (1.8, 3.73) and (2.5, 4.26) are used as initial cluster centers for K-Means algorithm.

A. Algorithm for Proposed approach

Input: D= {d1, d2, d3…..dn}
K= No. of cluster

Output: K Clusters

Procedure:

1. Initialize the OpenMP Processes= {P1, P2,.. Pn}.

2. Take the value of K as given by user.

3. Master process calculates K centroids as follows and broadcast it to other process

3.1 Generate the range of the initial centroids using following:

$$Mi = \max(Dj) - \min(Dj)/K$$
Where j= 1,2……n

3.2 Obtain the initial cluster centers Ck using the following equation:

$$Ck = \min(Dj) + (K-1) * M$$

4. For each process **repeat** steps a-c;

a. Calculate the Euclidean distance as similarity measure of each attribute Dj

Dist. = min (|| Dj – Ck ||2)1/2.

b. Assign data points to the clusters having minimum distance from the centroid.
c. Calculate new mean of each cluster

Until convergence criterion is met.

7. Master process gathers the all cluster results by slave processes.
8. Mater process declares the clustering results

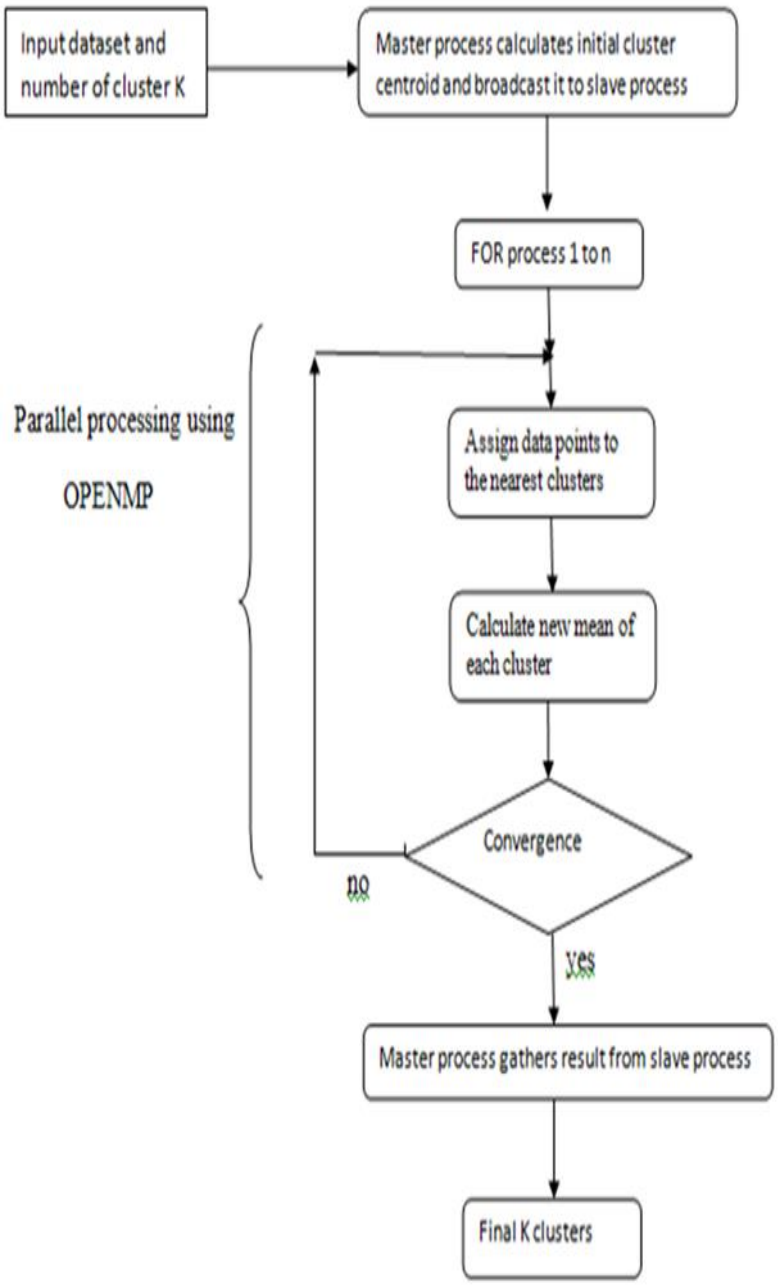

Fig. 1 Flowchart of Proposed approach.

The above approach can be used to reduce the execution time of K-means clustering

V. EXPERIMENTAL RESULTS

In this section, we first give the experimental environment in Section *A*. Second, we give the description of experimental data sets in Section *B*. Last, we give the experimental results and analysis in Section *C*.

A. Experimental environment

The hardware platform in this paper uses a PC with the configuration: Intel (R) Core (TM) Duo CPU @ 2.2 GHZ, 3 GB RAM and 320 GB hard disk. The software environment uses the following configuration: Ubuntu 14.04 operating system and GCC compiler; the parallel environment is the Linux version of OpenMP standard. The OpenMP API is present in GCC compiler.

B. Data sets

All experimental data sets are selected from the UCI Machine Learning Dataset Repository [15]. The information of all data sets is illustrated as shown in Table 2. In this table, two testing data sets are listed corresponding to the number of instances.

TABLE 1: Dataset Information

Sr.no	Name of dataset	Number of instances
1	Twenty Newsgroup	20000
2	Soybean	307

C. Experimental results and analysis

In our experiments, the time cost is the key performance. The I/O time and clustering time are calculated respectively for serial K-means algorithm and parallel implementation of our proposed K-means algorithm. To reflect the fairness and authenticity of the proposed algorithms, the number of processes is two in parallel implementation of K-means algorithm. Table 2 and Table 3 reports the clustering time for the aforementioned two algorithms for different value of cluster.

TABLE 2: Experimental Results on Twenty newsgroups Dataset

Sr.no	Number of cluster	Serial K-Means Time Taken (Seconds)	Proposed Parallel K-Means Time Taken (Seconds)
1	5	1.3025	0.583
2	10	2.7261	1.0774
3	15	6.6654	2.4657
4	20	6.7000	2.6793

TABLE 3: Experimental Results on Soybean Dataset

Sr.no	Number of cluster	Serial K-Means Time Taken (Seconds)	Proposed Parallel K-Means Time Taken (Seconds)
1	5	0.019	0.007
2	10	0.031	0.020
3	15	0.044	0.028

VI .CONCLUSIONS AND FUTURE WORK

As the dataset size increases the efficiency of k-means algorithm decreases. The time complexity is depends on choosing the initial centroids. To overcome the difficulties, we have proposed parallel K-means algorithm that uses the initial cluster generation process and parallel processing using OpenMP to reduce time taken for clustering the data. Results have shown that time taken for clustering using our proposed approach is optimized as compared to serial K-Mean approach.

FUTURE WORK

The ongoing research and our future work will be on Implementation K-Means Algorithm by hybrid architecture using OpenMP, MPI and CUDA for more efficient purpose in term of time complexity.

VII. ACKNOWLEDGMENT

The efforts were guided and supported relatively by faculty of Computer Science and Engineering department, Rajiv Gandhi College of Engineering and Research and Yeshwantrao Chavan College of Engineering, Nagpur, India

REFERENCES

[1] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review", ACM Computing. Surveys, vol. 31, pp. 264-323, 1999.

[2] K A Abdul Nazeer et al., "Enhancing the k-means clustering algorithm by using a O(n logn) heuristic method for finding better initial centroids", IEEE Second International Conference on Emerging Applications of Information Technology,pp-261-264, 2011.

[3] J Macqueen, "Some methods for classification and analysis of multivariate observations". Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, 281--297, University of California Press, Berkeley, California, 1967.

[4] S. Z. Selim and M. A. Ismail, "K-means type algorithms: A generalized convergence theorem and characterization of local optimality", IEEE Transactions on Pattern Analysis and Machine Intelligence., vol. 6, pp. 81-87,(1984).

[5] A. K. Jain, "Data clustering: 50 years beyond K-means", Pattern Recognition Letters archive. Volume 31 Issue 8, June, 2010.

[6] Y. T. Kao, E. Zahara, and I. W. Kao, "A hybridized approach to data clustering, ", Expert Systems with Applications, vol. 34, no. 3, pp. 1754–1762, 2008.

[7] Yugal Kumar and G. Sahoo, "A New Initialization Method to Originate Initial Cluster Centers for K-Means Algorithm", International Journal of Advanced Science and Technology Vol.62, 2014.

[8] DS.Bhupal Naik, S. Deva Kumar, S.V Ramakrishna, "Parallel Processing Of Enhanced K-Means Using OpenMP", IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[9] Margaret H. Dunham, "Data Mining- Introductory and Advanced Concepts", Pearson Education, 2006.

[10] Sanjay Goil, Harsha Nagesh, Alok Choudhary, "MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets" , 1999.

[11] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy: "Advances in Knowledge Discovery and Data Mining", American Association for Artificial Intelligence Press, 1996.

[12] Mitsuhisa Sato, "OpenMP: Parallel programming API for shared memory multiprocessors and on-chip multiprocessors", International Symposium on System Synthesis (ISSS) 2002, Kyoto, Japan, 2002.

[13] FANG Yuan, Zeng-Hui Meng, , Hong-Xia Zhanhz, Chun-Ru Dong , "A New Algorithm To Get the Initial Centroids", Third International Conference on Machine Laming and Cybernetics, Shanghai, 2004.

[14] Fahim A.M. Salem A.M. Torkey F.A. Ramadan M.A., "An efficient enhanced k-means clustering algorithm", Journal of Zhejiang University, 10(7):1626-1633, 2006.

[15] UCI Repository of Machine Learning Databases Available: http://archive.ics.uci.edu/ml/datasets/

[16] Abdolreza Hatamlou, "In search of optimal centroids on data clustering using a binary search algorithm", Pattern Recognition. Letter archive. vol. 33, pp. 1756-1760, 2012.

[17] Fahad. A, Ashtari.N, "A Survey of Clustering Algorithms for Big Data: Taxonomy & Empirical Analysis", IEEE Transactions on Emerging Topics in Computing, 2014.