COMPARATIVE ANALYSIS OF PARALLEL K MEANS AND PARALLEL FUZZY C MEANS CLUSTER ALGORITHMS

[1]Juby Mathew, [2]Dr. R Vijayakumar

Abstract: In this paper, we give a short review of recent developments in clustering. Clustering is the process of grouping of data, where the grouping is established by finding similarities between data based on their characteristics. Such groups are termed as Clusters. Clustering is a procedure to organizing the objects into groups or clustered together, based on the principle of maximizing the intra-class similarity and minimizing the inter class similarity.

A comparative study of clustering algorithms across two different data tests is performed here. The performance of the Parallel k means and parallel fuzzy c means clustering algorithms is compared based upon two metrics. One is an evaluation based on the execution time and the other is on classification error percentage and efficiency. After the experiment on two different data sets, it is concluded that both the algorithms performed well but the computational time of parallel K means is comparatively better than the parallel FCM algorithm. In both sequential and parallel computing FCM performs well but in parallel processing the execution time is considerably decrease compared with Parallel K means.

Keywords: ***Clustering, k-means, parallel k-means, fuzzy c means, parallel fuzzy c means***

I. INTRODUCTION

In the computing world, sequential computing is the use of a single processing unit to process a single or multiple tasks, and this has historically been the standard mode of computation. In contrast, parallel computing makes use of more than one central processing unit at the same time in order to allow users to complete lengthy computational tasks more quickly. Parallel computing differs from multi tasking where a single processor gives the appearance of working on two (or more)tasks by splitting its time between programs; if both the programs are computationally intensive then it will take more than twice the time for them to complete. It is designed to explore an inherent natural structure of the data objects, where objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible. The equivalence classes induced by the clusters provide a means for generalizing over the data objects and their features. Clustering methods are applied in many domains, such as medical research, psychology, economics and pattern recognition. [1]

Clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. There are huge amount of algorithms that have been developed for clustering large number of objects. These algorithms have different basic approaches to the problem, that can be categorizes as follows: (i) partition-based algorithms, (ii) hierarchical algorithms, (iii) density-based algorithms, (iv) grid-based algorithms (v) model-based algorithms and (vi) fuzzy algorithms. These algorithms are very different from each other in many aspects, for example some of them can handle noise while other do not care about it. In some cases the distance of the objects is important, in other cases the density or the distribution of the objects is an essential aspect. For this reason, comparing the different methods is a difficult and challenging task.

II. CHALLENGES OF CLUSTERING

Since there are various algorithms available for the clustering, it is very difficult to choose the suitable algorithm for a particular dataset. There are many limitations in the existing algorithms; some algorithms have problems when the clusters are of differing sizes, densities, having non-globular shapes. Some algorithms are sensitive to noise and outliers. Each algorithm has its own run time, complexity, error frequency etc. Another issue may be that the outcome of a clustering algorithm mainly depends on the type of dataset used. With the increase in the size and dimensions of dataset, it becomes difficult to handle for a particular clustering algorithm. The complexity of data set increases with the inclusion of data like audios, videos, pictures and other multimedia data which form very heavy database. There is a great need to choose the efficient clustering algorithm for the dataset. The selection of a clustering algorithm may based on the type of dataset, time requirement, efficiency needed, accuracy required, error tolerance etc. so the main challenge is to choose the correct type of clustering algorithm to get the desired results.[2]

We implement our extension of the Parallel k-means algorithm using Fork/Join method in JAVA.

Fork/Join parallelism [8] is a style of parallel programming useful for exploiting the parallelism inherent in divide and conquer algorithms, taking the typical form:

```
if (my portion of the work is small enough)
  do the work directly
else
{
   Split problem into independent parts
   Fork new subtasks to solve each part
   Join all subtasks
}
```

Fork-join executor framework has been created which is responsible for creating one new task object which is again responsible for creating new sub-task object and

waiting for sub-task to be completed. Internally it maintains a thread pool and executor assign pending task to this thread pool to complete when one task is waiting for another task to complete. The rest of the paper is organized as follows. Section 3 describes related work. Section 4 shows experimental results and evaluations. Finally, the conclusions and future work are presented in Section 5

III. RELATED WORK

A serial k-means algorithm was proposed in 1967 and since then it has gained great interest from data analysts and computer scientists. The algorithm has been applied to variety of applications ranging from medical informatics, genome analysis, image processing and segmentation, to aspect mining in software design. Despite its simplicity and great success, the k-means algorithm is known to degrade when the dataset grows larger in terms of number of objects and dimension. To obtain acceptable computational speed on huge datasets, most researchers turn to parallelizing scheme.

A. K-MEANS CLUSTERING

K-Means is a commonly used clustering algorithm used for data mining. Clustering is a means of arranging n data points into k clusters where each cluster has maximal similarity as defined by an objective function. Each point may only belong to one cluster, and the union of all clusters contains all n points. The algorithm assigns each point to the cluster whose center (also called centroids) is nearest. The center is the average of all the points in the cluster that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.[4] The algorithm steps are:

1) Choose the number of clusters, k.
2) Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers.
3) Assign each point to the nearest cluster center.
4) Re-compute the new cluster centers.
5) Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed).

The main disadvantage of this algorithm is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. It minimizes intra-cluster variance, but does not ensure that the result has a global minimum of variance. If, however, the initial cluster assignments are heuristically chosen to be around the final point, one can expect convergence to the correct values.

B. PARALLEL K MEANS ALGORITHM

Parallel K means consists of parallelizing the first phase of each step, where we calculate the nearest centroid to each point. Indeed, since k is generally a few orders of magnitude larger than the amount of cores currently available, the algorithm is bound by this phase. The algorithm computes the Euclidean distance of each point to the chosen set of centroids and tries to reassign each point to the nearest cluster. All the initial cluster centroids, calculating its assigned data point's distance from all of them, finding the minimum distance from its data point to a centroid, and become a member of the cluster represented by the centroid. When all threads are finished doing this, the membership of each data point is known, and thus the first phase of the algorithm is finished. If the number of processing elements were equal to the number of data points, this pass could finish in one step. For the next step of the algorithm where centroids of clusters are recalculated from the recently assigned member points, the same approach as the serial algorithm is utilized.[5]

Pseudo code for parallel k means

Input: a set of data points and the number of clusters, K
Output: K centroids and members of each cluster
Steps
1. Set initial global centroid C = <C1, C2, ..., CK>
2. Partition data to P subgroups, each subgroup has equal size
3. for each P,
4. Create a new process
5. Send C to the created process for calculating distances and assigning cluster members
6. Receive cluster members of K clusters from P processes
7. Re calculate new centroid C"
8. If difference(C, C")
9. Then set C to be C" and go back to step 2
10. Else stop and return C as well as cluster members

ADVANTAGES

1. For large number of variables, K Means algorithm may be faster than hierarchical clustering, when k is small.
2. K-Means may produce constricted clusters than hierarchical clustering, if the clusters are globular.

DISADVANTAGES

1. Difficulty in comparing quality of the clusters formed.
2. Fixed number of clusters can make it difficult to forecast what K should be.
3. Does not give good result with non-globular clusters. Different primary partitions can result in different final clusters.
4. Different initial partitions can result in different final clusters.[8]

C. FUZZY C MEANS CLUSTERING

Fuzzy C-means clustering (FCM), relies on the basic idea of Hard C-means clustering (HCM), with the difference that in FCM each data point belongs to a cluster to a degree of membership grade, while in HCM every data point either belongs to a certain cluster or not. So FCM employs fuzzy partitioning such that a given data point can belong to several groups with the degree of belongingness specified by membership grades between 0 and 1. However, FCM still uses a cost function that is to be minimized while trying to partition the data set.[6]

The general fuzzy inference process proceeds with the following steps:

1. The first step is called the FUZZIFICATION. In this step the membership functions defined on the input variables are applied to their actual values, to determine the degree of truth for each rule premise.
2. The second step is called the INFERENCE. The truth value for the premise of each rule is computed in this step and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Usually minimum or the product is used as inference rules
3. The third step is the COMPOSITION, in which all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable.
4. Finally DEFUZZIFICATION is performed to convert the fuzzy output set to a crisp number.

FCM algorithm is one of the most important fuzzy clustering methods, initially proposed by Dunn , and then generalized by Bezdek.[6][7] FCM algorithm is a technique of clustering which permits one piece of data to belong to two or more clusters. The aim of the FCM algorithm is the assignment of data points into clusters with varying degrees of membership values. Membership values lie between 0 and 1. This membership value reflects the degree to which the point is more representative of one cluster than the other.[10]

This algorithm works by assigning membership to each data point corresponding to each cluster center on the basis of distance between the cluster center and the data point. More the data is near to the cluster center more is its membership towards the particular cluster center. Clearly, summation of membership of each data point should be equal to one. After each iteration membership and cluster centers are updated according to the formula:

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/m-1)} \quad (1)$$

$$v_j = (\sum_{i=1}^{n} (\mu_{ij})^m x_i) / (\sum_{i=1}^{n} (\mu_{ij})^m), \forall j = 1, 2, ...c \quad (2)$$

where, *'n'* is the number of data points. *'vj'* represents the j^{th} cluster center. *'m'* is the fuzziness index m ∈ [1, ∞]. *'c'* represents the number of cluster center. *'μij'* represents the membership of i^{th} data to j^{th} cluster center. *'dij'* represents the Euclidean distance between i^{th} data and j^{th} cluster center.

Main objective of fuzzy c-means algorithm is to minimize:

$$J(U,V) = \sum_{i=1}^{n} \sum_{j=1}^{c} (\mu_{ij})^m \left\| x_i - v_j \right\|^2 \quad (3)$$

where, $'||x_i - v_j||'$ is the Euclidean distance between i^{th} data and j^{th} cluster center.

D. PARALLEL FUZZY C MEANS

The parallel FCM cluster analysis procedure is described by the following sequence:

Step 1: Splits the data set equally among the available processors so that each one receives N/ p records, where N is the number of records and p is the number of processes

Step 2: Compute the geometrical center of its local data and communicate this center to all processors, so that every processor can compute the geometrical center of the entire database.

Step 3: Sets initial centers and broadcasts them, so that all processors have the same clusters' centers values at the beginning of the FCM looping.

Step 4: Until convergence is achieved compute the distances from each record in the local dataset to all clusters' centers; update the partition matrix and calculate new clusters' centers.

Step 5: If the range of number of clusters is covered, stops, otherwise returns to Step3.

The procedure described above is computed for each number of clusters in the cluster analysis, so that the procedure is repeated as many times as the desired range of numbers of clusters [11]

ADVANTAGES

1) Gives best result for overlapped data set and comparatively better then k-means algorithm.
2) Unlike k-means where data point must exclusively belong to one cluster center here data point is assigned membership to each cluster center as a result of which data point may belong to more than one cluster center.

DISADVANTAGES

1) Apriori specification of the number of clusters.
2) With lower value of β we get the better result but at the expense of more number of iteration.
3) Euclidean distance measures can unequally weight underlying factors.

The clustering algorithms are compared according to the following factors

a) Size of dataset
b) Number of clusters
c) Time taken to form clusters

The clustering algorithms are divided into two categories partitioning based and non partitioning based. Firstly partitioning based clustering algorithms and non partitioning based algorithms are compared separately and the results have been drawn. Then the partitioning and non partitioning based algorithms are compared

IV. EXPERIMENTAL RESULTS

We implemented the proposed algorithm, parallel k-means and the Parallel fuzzy c means using JAVA language. The code is executed on dell inspiron N4030 Laptop, Intel(R) Core(TM) i5 Processor 2.67 GHz, 2 MB cache memory, 3GB RAM, 64-bit Windows 7 Home and Netbeans IDE 8.0. We evaluate performances of the two algorithms on two-dimensional dataset, four clusters. The computational time of k-means as compared with fuzzy c means is given in Table 1 and computational speed of parallel k-means and parallel fuzzy c means is given in Table 2.To analyze

two tables data, it reveals that compared with sequential processing parallel model would be better execution time. Running time comparison of K means and FCM is graphically shown in Figure 2 and running time comparison of Parallel K means and Parallel FCM is graphically shown in Figure 3.Computational time of Parallel fuzzy c means greater than parallel k means.

Table 1: The execution time of k means and Fuzzy c means

Dataset Size N	Exec.Sec KM	Exec.Sec FCM	Time diff.
100K	10.12	10.12	0
200K	18.56	18.95	0.39
1000K	81.8	120.4	38.6
2000K	165.2	215.9	50.7
4000K	385.4	475.8	90.4

Table 2: The execution time of Parallel k means and Parallel Fuzzy c means

Dataset Size N	Exec.Sec PKM-	Exec.Sec PFCM	Time Diff
100K	3.12	3.12	0
200K	6.56	6.57	0.01
1000K	31.8	40.4	8.6
2000K	65.9	85.2	19.6
4000K	115.1	145.7	30.6

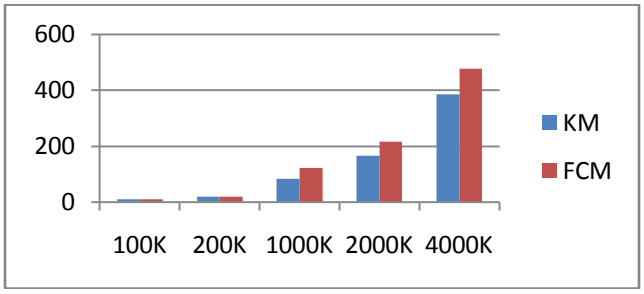


Fig.1 Running time Comparison KM & FCM

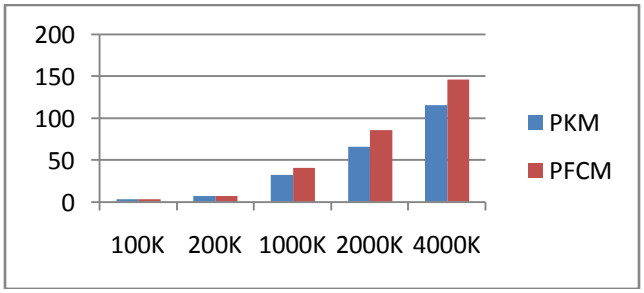


Fig.2 Running time Comparison PKM & PFCM

PERFORMANCE ANALYSIS

Performance of the proposed algorithm is evaluated by classification error percentage (CEP) and classification efficiency. This method is successfully applied by Senthilnath.[12]

Classification Error Percentage (CEP)

The classification of each pattern is done by assigning it to the class whose distance is closest to the center of the clusters. Then, the classified output is compared with the desired output and if they are not exactly the same, the pattern is separated as misclassified. Let **n** be the total number of elements in the dataset and **m** be the number of elements misclassified after finding out the cluster center using the above algorithms. Then classification error percentage is given by

$$CEP = \frac{m}{n} * 100 \qquad (4)$$

Classification efficiency

Classification efficiency is obtained using both the training (75%) and test data(25%). The efficiency is indicated by the percentage classification which tells us how many samples belonging to a particular class have been correctly classified**.** The percentage classification (μi) for the class ci

$$\mu i = \frac{qii}{\sum_{j=1}^{n} qji} \qquad (5)$$

Where $\mathbf{q_{ii}}$ is the number of correctly classified samples and **n** is the number of samples for the class **ci** in the data set**.**

We use two dataset for calculating CEP and efficiency for the two algorithms. The Iris data set consists of three varieties of flowers. There are 100 instances and 4 attributes that make up the 4 classes. The Wine data obtained from the chemical analysis of wines were derived from three different cultivators. The data set contains 4 types of wines, with 100 patterns and 4 attributes. Table 3 shows that classification error percentage of two data set like Iris and Wine using eq.4.Table 4 shows that Classification efficiency of two data sets calculated by eq.5

Table 3: Classification error percentage

	PKM	PFCM
Iris	0.002	0
Wine	1.23	0.10

Table 4: Classification efficiency

	PKM(%)	PFCM (%)
Iris	95	100
Wine	90	99

The above results shows that Parallel fuzzy c means algorithms is efficient in compared with parallel k means algorithm.

V. CONCLUSION

We presented a short review about modern trends in clustering .Obviously this short review cannot be complete, referring to the cited literature for further reading. A comparative study of clustering algorithms across two different data sets is performed. After the experimental studies on two different data sets, it is

concluded that both the algorithms performed well but the computational time of parallel K means is comparatively better than the parallel FCM algorithm. In both sequential and parallel computing FCM performs well but in parallel processing the execution time is considerably decrease compared with Parallel K means. In classification error percentage shows that parallel FCM are the best solution for generating optimal cluster centers. The performance measure using classification efficiency of the Parallel FCM is analyzed using two problems. From the results obtained, we can conclude that the Parallel FCM is an efficient, reliable method, which can be applied successfully to generate optimal cluster centers.

REFERENCES

[1].Han, J., Kamber, M. (eds.): Data Mining Concepts and Techniques, 2nd Elsevier, San Fransisco (2006)

[2]P.Berkhin, 2002, Survey of Clustering Data mining Techiniques.Techinical report, Accrue Software, San Jose.

[3]. L.Hall, B.Ozyurt, and J.Bezdek,"Clustering with a Genetically Optimized Approach,"IEEE Transactions on Evolutionary computation", vol.3, No.2, 1999.

[4]. K.A Abdul Nazeer, M.P Sebastian "Improving the Accuracy and Efficiency of the K-means Clustering Algorithm "WCE 2009, London.

[5]. Li X. and Fang Z., "Parallel clustering algorithms", Parallel Computing, 1989, 11(3): pp. 275-290.

[6] Dunn J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, vol. 3, pp. 32–57, 1974.

[7] Bezdek J C. Pattern recognition with fuzzy objective function algorithms, Plenum Press, New York, 1981.

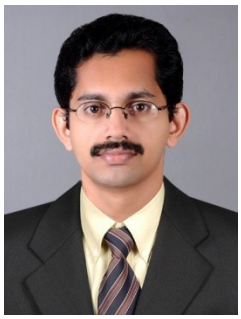
[8]. Zhang Y., Xiong Z., Mao J., and Ou L., "The study of parallel k-means algorithm", Proceedings of the 6th World Congress on Intelligent Control and Automation, 2006, pp. 5868-5871.

[9]. Doug Lea, A Java Fork/Join Framework, State University of New York at Oswego

[10] Xiaojun LOU, Junying LI and Haitao LIU. Improved fuzzy C-means clustering algorithm based on cluster density. Journal of Computational Information Systems, 8(2), pp. 727–737, 2012.

[11] Jie Lio, Chao-Hsien Chu, Wang and Yungeng. An In-depth analysis of fuzzy c-means clustering for cellular manufacturing.
In: Fifth international conference on fuzzy systems and knowledge discovery (FSKD'08), pp. 42-46, 2008.

[12]. J. Senthilnath, S.N. Omkar, V. Mani,
Clustering using firefly algorithm: Performance study, Swarm and Evolutionary Computation 1 (2011) 164–171



Juby Mathew pursued his MCA from Periyar University Salem and MPhil in Computer Science from Madurai Kamaraj University and MTech from M S University Tirunelveli.Currently he is pursuing his PhD in parallel algorithms from school of Computer science, Mahatma Gandhi University Kottayam, Kerala, India. So far he has published five international journals and presented papers in more than ten National and International conferences. He has more than ten years of teaching and corporate experience and continuing his profession in teaching. Now he is working as an Assistant Professor in MCA Department of AmalJyothi College of Engineering Kanjirapally, Kerala, India.

