International Journal of Computer Science and Business Informatics

IJCSBI.ORG

Scalable Rough C-Means clustering using Firefly algorithm

Abhilash Namdev
School of Computer Science and Engineering
VIT University, Vellore - 632014, India
E-mail: abhilash.namdev8@gmail.com

B.K. Tripathy
School of Computer Science and Engineering
VIT University, Vellore - 632014, India
E-mail: tripathybk@vit.ac.in

ABSTRACT

Our main interest is in dealing with the disadvantages of old clustering algorithms and coming up with a method that can generate clusters which produce optimal results when compared with previous approaches. Firstly, our focus is on analyzing the limitations of most widely used clustering algorithm. Here we choose K means clustering algorithm for the purpose. To provide the optimal results from the initial stage of algorithm we use firefly algorithm. The bioinspired algorithm that generates optimal minimum or maximum values based on certain parameters. To avoid the strictness on the boundary area in k means algorithm, we choose Rough C means algorithm, which provide some flexibility during the clustering process. Our proposed method provides most efficiency both in terms of time and space. We used efficient data structures which help us to avoid waste of memory while computation and also our algorithm utilizes maximum resources of the machine to make the execution rate as fast as possible.

Keywords

Clustering, datasets, firefly, threads, k-means, rough c mean

1. INTRODUCTION

The process of clustering is the organization of patterns into sensible groups which helps us to find the similarities and dissimilarities among the patterns and to derive conclusion about them. We can easily find such type of grouping in the fields of medical science, geography, biology, engineering and sociology. This process of grouping comes under the unsupervised learning. The basic steps involve in this grouping task starts with feature selection, and followed by proximity measures, clustering criterion, clustering algorithm, then validation of results and interpretation of results come under the process. There are many directions where cluster analysis is in use, some of them are data reduction, hypothesis generation, hypothesis testing and prediction based on cluster.

To understand the fundamentals of any clustering process, let's take any dataset named X. Now k-clustering defines that the partition of dataset X into k groups $G_1, G_2, ...G_k$ such that the following conditions must be satisfy

- $G_i \neq \phi; i = 1, 2, ...k$
- $\bigcup_{i=1}^{k} G_i = X$
- $G_i \bigcap G_j = \phi; i \neq j \wedge i, j = 1, 2, ...k$

This type of clustering is sometimes called hard or crisp. The alternative of above clustering are fuzzy and rough type of clustering, where element from the dataset X can belong to more than one clusters. One of the most popular and well known examples of hard clustering algorithmic scheme is k means or c means algorithm. Here squared Euclidian distance is used to find the dissimilarity among the elements of dataset. However, the results of this version of algorithm highly depend on the order in which initial centroids are chosen. (McCullock John, 2012) The main advantage of c-means or k-means algorithm is its simplicity in computing the process. But k means is not suitable for the categorical data i.e., it results well only with continuous data valued.

Lingras proposed a clustering approach which was based on the concept of rough set theory, known as Rough c means algorithm. This algorithm describes the group by the value of its centroids as well as upper and lower approximation. Rough c means algorithm solve the problem of k means algorithm to some extent by allowing element to be the part of either one cluster or between two clusters. Elements in the lower approximation are considered to be completely belonging to that class of elements.

The problem with both the above discussed algorithm is that they both are sensitive to the initial cluster-centroids. Since initial centroids are chosen randomly, many times the algorithm doesn't produce optimal results. Therefore, in the proposed system, we have used firefly algorithm to determine the values for the initial class centroids. Then these optimal values are applied to rough c-means algorithm, which ultimately increase the accuracy of cluster. The firefly algorithm is implemented on both these clustering approaches and the results are comparatively analyzed. The idea behind the firefly algorithm work around the flashing behavior of the fireflies. Each firefly has its degree of attractiveness and intensity of light. Based on these parameters the firefly is attracted toward other firefly in the working space. On the basis of this brightness parameter, the firefly with low brightness takes a moment towards the firefly high brightness value at each of the iterations. And the values of this parameters update on usual period. The best position of the firefly is selected after running for the specific period of time. These optimal values are considered as the centroid values for our algorithm.

The system is designed in java technology, which is one of the most efficient and widely used high-level object oriented language of the world. The implementation in java is very easy, manageable and understandable by the programmer. To make the system efficient and flexible many dynamic approaches has been adopted in the side of data structure. Data structures are the arrangement of the data inside the computer's memory. And algorithm ultimately modifies this structure in various ways. Data structures are chosen to make the system efficient in terms of both space and time; Like Arraylist acts as dynamic array which can grow or shrink as per the requirement. This ultimately avoids the wastage of memory as compared to normal array concept, where size of the array is fixed in advance. Which lead to the waste of memory when the part of memory doesn't even used during the computations. Another measure adopted in order to utilize the waste of time during the user interaction, which is the concept of multithreading. Java provides the built in support for the multithreaded programming. The multithreaded program contains more the one part that can work simultaneously. (Singh Chaitanya, 2016) Multithreading provide us the facility to write efficient program that utilize the maximum processing power. Since Multithreading is important for interactive systems, we have used it in our proposed system. To time spent by the user to interact with the system, this time is in background utilize to run the basic operations of clustering. The task performed in the background is usually not dependent on the activity of user.

The system is tested over different datasets collected from the dataset repository available on the internet. At the present stage all the dataset used for the experiments are in the excel format, since Microsoft excel is the one

which is most widely used for storing data across the world. The datasets consist of the large scale and high dimensional data. The number of instances and attributes of the datasets are varied from 5 to 5000 in numbers.

2. REVIEW OF LITERATURE

In 1967, J.B MacQueen has first proposed the K means algorithm (MacQueen, 1967), which is now the most widely used clustering algorithm in the field of data mining. (Mathew et al, 2014a) have introduced the concept of parallel clustering approach of K-means algorithm. They found in (Mathew et al, 2014a) that average number of iterations the program takes was lower than the old approaches. In (Swamy et al, 2015) it is concluded that as the size of the data increases, the time taken by K-means algorithms also increases with high rate. They (Swamy et al, 2015) suggested the concept of parallel processing was introduced to decrease the execution time during the clustering process. Then Yang (Yang, 2010) has introduced the concept of firefly in 2008. According to them (Swamy et al, 2015) firefly algorithm is the swarm intelligence algorithm, inspired by insects with their unique property of flashing. In (Raja et al, 2013) the various randomization parameters are analyzed and the conclusion that the value which is best suited on respective condition are provided. In (Lohrer et al, 2013) it is suggested that firefly algorithm is a very efficient algorithm that achieves optimal results when compared with Particle swarm optimization (PSO). Also time taken to execute this algorithm is very less as compared to other approaches. They (Lohrer et al, 2013) are still working on the area to develop firefly on the hardware of system, which leads to even lesser time as compared to current approach.

In (Zhang et al, 2006) the concept of dynamic load balancing is introduced. They also suggested the use of parallel programming to implement k means algorithm and step up the efficiency by using load balancing between the core of the machine. They have adopted the strategy of master/slave design model. (Karinor et al, 2015) tried two different approaches for k means algorithm. One is for small scale dataset and other is for large scale data using both serial and MapReduce implementation of parallel algorithm. And concluded that with the increase in iterations of algorithm, the computation overload also increases, i.e., they both are directly proportional. In (Aamir et al, 2014) it is explained as how to use parallel programming in java. This is the fact that if we want to enhance the execution rate of our program then we must have to learn the concept of parallel programming. They have explained very well as how parallel programming tools and techniques are used in both shared and distributed environment.

(Chen et al, 2011) studied the performance issues of java program using multithreading on multicore machine. They examined the tuning of JVM in their research. To utilize the maximum benefits of multithreading concept we must have to efficiently utilize the cache memory. Firefly algorithm can give its best results only when the best parameters and objective functions are selected. (MO et al, 2013) closely examined the effect of each parameter in firefly algorithm. They suggested certain guidelines in determining the values of these parameters. They used different functions that test the effect on fireflies by parameter change. Since proper start is important for K-means algorithm to achieve optimal results, (Xu et al, 2014) have proposed K-means++ algorithm. This algorithm increases the productivity of standard k means approach by using Map reduce technique. They successfully tested their approach on both real and synthetic data.

FIREFLY ALGORITHM

Inspired by the flashing property of fireflies, in (Yang, 2010) the firefly algorithm is proposed. Every firefly is having its unique property with respect to light and fitness. Attractiveness is directly proportional to light intensity.

a. Fireflies are also known as agents and let x_i be the position of the i^{th} firefly in d dimension.

b. Initial fitness values for all the fireflies can be set by using

$$F(X) = \sum_{i=0}^{D} X^2 \qquad ; X \rightarrow xi$$

c. The distance r_{ij} between the two fireflies, say the i^{th} and j^{th} ones can be calculated as

$$r_{ij} = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2}$$

d. The firefly with lower brightness values is attracted towards the firefly with higher brightness value.

e. In that case the position update of firefly can be calculated by

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \beta_0 / \left(1 + \gamma r_{ij}^2\right) \left(\boldsymbol{x}_j - \boldsymbol{x}_i\right) + \alpha S(R - 0.5)$$

Where β is degree of attractiveness and α is a random parameter.

f. After that, light intensity is updated using the formula

$$I(r) = I_0 e^{-\gamma d^2}$$

Where I_0 is initial intensity, and γ is absorption coefficient.

g. Arrange the firefly in order and find the best firefly.

The position of theses optimal fireflies is considered to be the optimal values. These values are selected as the optimal centroid values for our system.

3. PROPOSED METHOD

After studying the previous approaches, we got the first thing to design good clustering algorithm is to select the optimal centroid values at the start. Also we have seen that the firefly algorithm is best suited when there is the need of optimal values. Hence, in our proposed method, we used the firefly algorithm to get the optimal initial centroid values at the early stage of the algorithm. To provide some sort of flexibility in the process of clustering, we implemented firefly algorithm on Rough C Means algorithms. Since we studied that K means algorithm is crisp or hard in the boundary region, we tried some new approach which is not tested yet. Rough C means algorithm allow the element to become the part of either completely in one class or in between two classes. This approach is used to make the clustering flexible.

Our proposed method is to implement Rough C means algorithm and provide the optimal result to it by adding the firefly algorithm. The results of this approach are tested on different set of data. The experiment was performed on a dataset whose number of instances varies from 5 to 5000. Mostly the dataset used for the experiment is the real numbered data, because it is quite easy to perform and handle real data for the purpose of experiment. The dataset is collected from the online repository available on the internet. The system is developed in JAVA technology because it is easy to program and highly manageable as compared to other programming language. (Oracle-Java-Documentation, 2015) The efficient data structures like Arraylist and concept of multithreading is used that allow the system to avoid wastage of time and space. The approach has been followed to make the code run on multiple processors. This can make the system to run faster than other approaches. The proposed method is then compared with existing method in terms of execution time.

Pseudo-code for proposed method

1. Start
2. Ask user to enter number of clusters needed → noc

3. Ask user to view his dataset
4. If yes, then display the dataset file
5. Call store() method, to store the data in temporary storage, to avoid accidental modification
6. Call firefly() method to initialize the value of centroids
   a. set the parameters α, β, ϒ
   b. calculate initial fitness values of each firefly → I[i]
   c. calculate distance between each two fireflies → r[i][j]
   d. if (I[i] < I[j])
   e. then firefly i move toward firefly j
   f. Update β and fitness value I
   g. Repeat until maxGen < noc
   h. Sort the fireflies and select best positions
7. Calculate the minimum distance of element from centroid of cluster
8. Calculate next to minimum distance of element from another centroid
9. Check if the difference between both the distances is less the threshold
10. If yes, the element belongs to boundary
11. Else, Assign elements to respective cluster
12. Recalculate mean based of each cluster and boundary elements
13. Repeat from step 7 until new centroid values are not equal to old centroid values
14. Display the final results by printing elements with respective cluster
15. Display, number of iterations algorithm took

The important function during the program execution carried out from the calculation of Euclidean distances in each iteration. The distance of the element from every centroid is computed. Then the minimum and second to minimum distance is collected for the use. If the difference between these two distances is less than some threshold value, then that particular element is considered to be the part of boundary between those two clusters. And other are considered being the part of lower approximation of that cluster. The system diagram of proposed method is shown next.

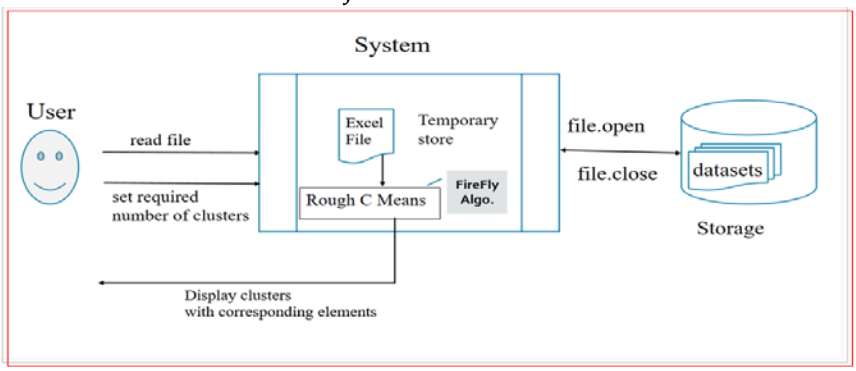

Fig.1. Model for proposed system

The figure clearly shows the interaction of User with the system. User is allowed to ask for his required number of clusters and he can also see his dataset on the console screen of the system. The diagram also projects the interaction of system with the secondary storage. User has to specify the address of his dataset in advance before performing clustering operation. The data from the file is collected in some temporary storage, and all the further operations are carried out from that data. This is done to avoid the accidental modification on original dataset during the operation. At this stage the system can only able to read the datasets from excel file. Since most of the datasets use Microsoft excel to store and manage data, hence our system in its early state is developed to work with only excel worksheets.

4. EXPERIMENTAL RESULTS

We have implemented our proposed system in java technology, because program in java is easy to manage. The execution of code is done on Intel(R) core(TM) i3 CPU with 2.53 GHz processor, installed memory (RAM) is 4.00 GB, while the system has 64-bit operating system (Windows 10).

Dataset description

Dataset 1: The voter dataset is used which has two attributes. The one represents age and another represents sex of the candidate. It has 439 numbers of instances. The data in the file is of type integer. While the age with value 1 represents female candidates and value 2 represents male candidate. Data with NULL values are left blank which is ignored by the algorithm.

Dataset 2: The weather dataset, which has 4 attributes named by city name, Year, Lowest temperature and highest temperature. The dataset has 3000

number of tuples or instances. The type of the data in the dataset is integer. In this dataset Null values are treated as zero.

Dataset 3: The third dataset is also weather dataset but with different size. Here the number of attributes is 9 named by order country, city, year, low T, high T, warm, cold, and average. This dataset has 5000 instances. This is the largest dataset we have tested on our system.

In addition to the technology, we refer certain tools in the process of development of the system. The program is developed in NetBeans IDE 8.0.2 because using NetBeans IDE is we can easily develop java desktop, mobile and web applications. It provides fast and smart coding environment for the developer. Another tool used for analyzing the outcomes and comparing with proposed work is WEKA. The WEKA is the tool for performing experiments with various algorithms from the data mining field. It is an open source software. It provides various datasets of different size for performing experiments. It also provides various visualization techniques for better understanding the results like graphs, tree etc.

To make it easily understandable, we are presenting the screen of outcomes produced while experimenting. Here we took the dataset with 5 instance and 2 attributes.

	X	Y
A	1	1
B	2	1
C	4	3
D	5	4

Fig.2. Sample dataset for demonstration

Optimal Centroid values collected using the firefly algorithm

```
********> Using FirFly Algorithm <********
Optimal Centroid values collected are :

 5.0      5.0
 3.751    3.046
 4.0      3.0
```

Fig.3. Results of Firefly algorithm

There are certain parameters, which are very carefully selected from previous researches. The sphere optimization function is adopted in firefly algorithm, because it easy to implement and mostly used for research. The other parameters like α, β, ϒ are put as constant values. Final results collected after running on Rough C means algorithm

```
********** Final Results are **********

Cluster 0 contains

        5.0     4.0     5.0     5.0
Cluster 1 contains

        1.0     1.0
Cluster 2 contains

        4.0     3.0

******** Elements on boundary area are ********


----> Boundary of Cluster 0 contains

# No Element present

----> Boundary of Cluster 1 contains

        2.0     1.0
----> Boundary of Cluster 2 contains

        2.0     1.0
```

Fig.4. Final Results after Clustering

On each run, the system will show the elapsed time. For example, the elapsed time of the above sample experiment is 16 milliseconds or 0.016 seconds. As given on the above results, the output of the algorithm appears in two categories. One with the elements belongs to lower approximation of the cluster and another with the element which are the part of boundary region. The performance of the system is also measured on the CPU scale. The number of cycles processor takes to execute the algorithm is also the point of concern. The frequency of our machine is about 2.48 GHz. The next chart taken during the time of execution, which primarily shows the basic details about the machine include cache size, memory, core and speed of processor. With the fundamental details, the chart clearly shows the amount of CPU utilization during the time of execution. We have calculated

97% of CPU utilization during the process of execution. The chart also displays the number of processes as well as number of threads running on the system.

The performance noted by task manage is shown below

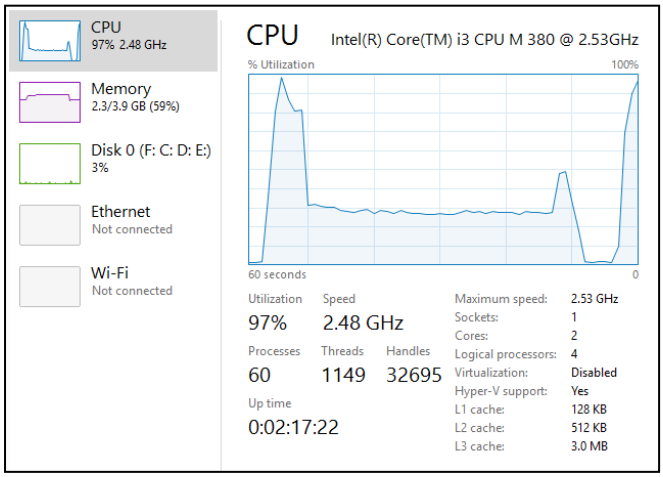

Fig.4. CPU utilization graph

The CPU usage in the above graph shows that processing power is utilizes up to 97 %. As discussed early, the experiment is carried out on different set of data collected from the repository available on Internet. The readings of the experiment are shown below by the table. The datasets are varied from different size in terms of both attribute and instances. The table below shows the execution time of different datasets when size of the dataset varies. And we found that as the size of the dataset increases, the execution time also increases. The execution time is measured in milliseconds. The first dataset covered in the table is same as the one shown previously.

Table 1: **Comparison of execution time**

Dataset	**No. of attributes**	**No. of instances**	**Execution time (milliseconds)**
Sample	2	5	16
Voter	2	439	832
Weather_1	4	3000	7953
Weather_2	7	5000	39298

The graph is plotted to understand the results with the mark. The points covered in the graph are execution time, number of instances and number of attributes. The execution time is measured with respect to different number of attributes and instances.

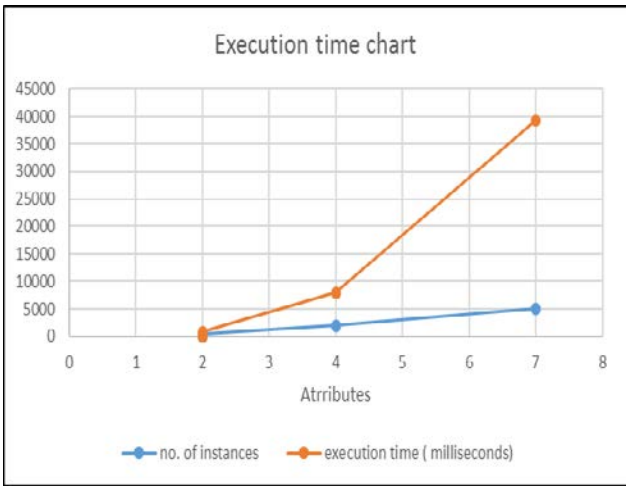

Fig.5. Execution time chart

The above chart shows that execution time is directly proportional to the increase in number of instances and attributes. The vertical axis is taken for time and horizontal axis is for number of attributes.

The selection of different parameters plays a vital role in the final results. Parameters like the threshold, alpha (α), beta (β), Gama (γ) are mostly taken from the previous researches performed on the selection of optimal parameters. Out of the above mentioned parameters, the threshold value of rough c-means generally varies with different dataset and user's requirements.

5. CONCLUSIONS

We observed that although the K-means algorithm for clustering is easy to implement and is the most widely used one, there are many drawbacks in this algorithm which pulls it away from resulting optimal output. Couple of main gaps in the algorithm are, first it is sensitive to the selection of initial centers for its cluster and another, that it doesn't permit any element to take a part of more than one cluster group. This motivated us to design something to solve this problem. So, we used the firefly algorithm, to get better start to the proposed clustering algorithm. After this we used

the rough c-means algorithm instead of the K means clustering, which provides more flexibility to the clustering phase by allowing elements to belong to more than one cluster. Our proposed method is successfully tested on datasets of different sizes. It is proposed to apply this approach by replacing the rough c-means algorithm with other algorithms and perform comparative analysis to find out the most suitable combination of firefly algorithm with clustering algorithm in this direction.

REFERENCES

Aamir, S. Akhtar, A. Javed, A. and Carpenter, B. (2014). *Teaching Parallel Programming Using Java*. Workshop on Education for High Performance Computing, pp.56-63.

Chen, K Y. Chang, J M. and Hou, T W. (2011). *Multithreading in Java: Performance and Scalability on Multicore Systems*. IEEE Transactions on Computers, pp.1521-1534.

Javier, F G,. (2012). *Java 7 Concurrency Cookbook*. Packet publications.

Karimov, J. Ozbayoglu, M. and Dogdu, E. (2015). *k-means Performance Improvements with Centroid Calculation Heuristics both for Serial and Parallel environments.* IEEE International Congress on Big Data, pp.444-452

Lohrer, M. F. (2013). *A Comparison Between the Firefly Algorithm and Particle Swarm Optimization.* Graduate Thesis, submitted to Oakland University

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations.* 5th Berkley Symposium, pp.281-297.

Mathew, J. and Vijayakumar, R. (2014a). *Scalable parallel clustering using modified Firefly algorithm.* IOSR Journal of Computer Engineering. Volume 16. Issue 6, Ver. I, pp.14-24.

Mathew, J. Vijayakumar, R. (2014b). *Scalable Parallel Clustering Approach for Large Data using Possibilistic Fuzzy C-Means Algorithm*. International Journal of Computer Applications. Volume 103. Number 9.

Mathew, J. and Vijayakumar, R. (2014). *Scalable Parallel Clustering Approach for Large Data Using Parallel K Means and Firefly Algorithms*. International Conference on High Performance Computing and Applications, pp.1-8.

MO, Y B. and MA, Y Z. (2013). *Optimal Choice of Parameters for Firefly Algorithm*, In: Fourth International Conference on digital manufacturing and automation (ICDMA). pp. 887-892.

Raja, M S M. Manic, K S. and Rajinikanth, V. (2013). *Firefly Algorithm with Various Randomization Parameters: An Analysis.* Springer International Publishing Switzerland, pp.110-121.

Scheldt, Herbert. (2011). *Java - the Complete Reference*. 8th edition. Mc-Graw Hill Companies. USA.

Swamy P, Raghuwanshi, M. and Gholghate, A. (2015). *An Improved approach for K-Means using Parallel Processing.* Proceedings of the 2015 International Conference on Computing, Communication, Control and Automation. pp.358-361.

Theodoratos, S. and Koutroumbas, K. *Pattern Recognition*, (2008). 4th edition, Academic Press.

Xu, Y. Qu, W. Li, Z. Min, G. Li, K. and Liu, Z. (2014). *Efficient k-Means++ Approximation with MapReduce*. IEEE Transactions on Parallel and Distributed Systems. pp. 3135 – 3144..

Yang, X.S. (2010). *Nature Inspired Metaheuristic Algorithms.* Luniver Press. BA11 6TT. UK.

Zhang, Y. Xiong, Z. Mao, J. and L, Ou. (2006). *The Study of Parallel K-Means Algorithm.* Proceedings of the 6th World Congress on Intelligent Control and Automation. Dalian, China, pp. 5868 – 5871

Singh, Chaitanya. (2016). Beginners Book for multithreading. [Online 1] available at: http://beginnersbook.com/ 2013/03/multithreading-in-java/.

Oracle Java Documentation. *(2015). Fork-join framework.* [Online 2] available at: https://docs.oracle.com/javase/tutorial/ essential/concurrency/ forkjoin.html.

McCullock, John.*(2012). Implementation of K-means clustering*. [Online 3] available at: http://mnemstudio.org/ clustering-k-means-introduction.htm.

This paper may be cited as:

Namdev, A. and Tripathy, B.K. 2016. Scalable Rough C-Means clustering using Firefly algorithm. *International Journal of Computer Science and Business Informatics, Vol. 16, No. 2, pp. 1-14.*