Hybrid Partition Around Medoids Algorithm for Large Volume of Data

Nway Yu Aung
University of Information Technology
nwayuaung@uit.edu.mm

Kyawt Kyawt San
University of Information Technology
kyawtkyawtsan@uit.edu.mm

Swe Zin Hlaing
University of Information Technology
swezin@uit.edu.mm

Abstract

Clustering is a crucial data-mining tool for analyzing valuable information from a massive data volume. Partition Around Medoids (PAM), one of the clustering algorithms that is simple, scalable and can easily implement but sensitive to initial medoids and vast amount of data. Meta-heuristics algorithms such as Ant Colony Optimization algorithm, Bat algorithm, Bees algorithm, etc. used to introduce the combinative in the clustering algorithm that will gives optimum medoids and hence find the better cluster quality. But, the main issue of very large data is in time consumption and lack of quality. To avoid issued of time consumption, existing clustering approaches are run on parallel frameworks. So, this paper proposed the hybrid approach to integrate PAM and Bat which one of meta-heuristic algorithm to obtain optimal initial medoids and PAM to get the better clusters. To handle a large number of datasets for fast and parallel processing, all experiments are done in Apache Spark Framework.

Key Words- Clustering, Bat algorithm, PAM algorithm, Apache Spark

1. Introduction

Clustering algorithms have been applied according to their different applications. Partition Around Medoids is one of the partition techniques of clustering. These are used in many application areas such as data-mining and health care, etc… Nowadays, various sources such as social media, internet of things, multimedia, sensor networks have been generated largest amount of data.

When confronted with large amount of data, PAM method has some drawbacks. For large volume of data, time consumption and lack of accuracy are main problem of this algorithm. So that the big challenge is the large volume of data by using PAM.

To overthrow the drawback of time consumption, many researchers have been proposed. Medoids based clustering algorithm is easy to implement but efficient. H.S.Park proposed a new medoids-based clustering algorithm. Just like means method, this algorithm runs a local heuristic by updating the medoids. This proposed method has reduced computation time than traditional PAM.

K-Medoids algorithm combined with parallelization techniques was proposed [2]. Every job has submitted to MapReduce procedures. In the map phase, each sample whose center is the most similar was assigned to one cluster. The intermediate center for each cluster calculated in the combine phase. The iterative procedure is stop when the center doesn't change. This method gets the good clustering result and also linear speedup.

K-Medoids parallel clustering algorithm is handle huge volume, variety and velocity of data based on MapReduce paradigm [3]. Another main point in the proposed algorithm is achieve parallelism independent of the number of k clusters to be formed. The proposed solution executed multiple computation nodes along independent of the k number of clusters to be discovered. Another drawback of the paper is the initial medoids. PAM algorithm chooses the initial cluster medoids by randomly. The cluster quality mainly depends on randomly choosing initial medoids and also different initial cluster medoids result in different cluster of PAM.

To overcome this problem, many methods have been put forward. Metaheuristic approaches such as ant colony optimization (ACO), particle swarm optimization (PSO), artificial bee colony (ABC) etc.… emerged promising tool for clustering [4][17][18]. Swarm Intelligence (SI), an artificial intelligence technique is depending on the collective property emerging from multi-agents in a swarm. SI has attracted the clustering community with the benefit of robust, scalable, and easily distributed. So, it has become a widely used. At the present time, combination of clustering and SI-based approach are rapidly emerged.

K-medoids clustering algorithm incorporated with PSO and the Simulated Annealing [5]. The experimental result shows enhances the speed of converge and accuracy. The clustering effect valuable than the *k* -medoids algorithm has been proved. R.Tiwari outlines an ACO algorithm applied in data mining fields [6]. The ants finding the shortest way for solving clustering and classification problems. The method show that enhance of the ACO applying the data clustering method and the best results.

This paper contributes to make medoids-based algorithm efficient for large dataset. To improve the time consuming, this paper implements the parallel algorithms on Apache Spark. To solve the initial medoids random problem, apply Bat algorithm to find optimal initial medoids. Rest of the paper is organized as, Section II give the theoretical background of the research work. Section III elaborates the proposed system. Section IV is experimental results. Finally, the paper is concluded in V section.

2. Theoretical Background

2.1. Partition Around Medoids (PAM)

The PAM algorithm is a popular of k-medoids clustering. The same as k-means algorithm, the initial medoids are chosen arbitrarily. This method improved the cluster quality by replacing a representative object with a nonrepresentative object. The iterative process continues until the medoids unchanged. The cost function which is the dissimilarity of the objects determined the cluster quality.

The procedures of PAM are as follow:

Step 1. Cluster the data into the clusters

Step 2. Select objects as medoids randomly

Step 3. Calculated the data object to its nearest medoids using distance method

Step 4. Calculate the total swapping cost for medoids and non-medoids items

Step 5. If swapping cost is less than 0, medoids is replaced by data object

Step 6. Go to step 2-5 until unchanged the medoids

The PAM algorithm is not sensitive for noisy data, but the time complexity of algorithm is very large, as

$$O(t^{*}k^{*}(n-k)2), \text{ where}$$

$t =$ the number of iterations,

$n =$ the total number of data set,

$k =$ the number of clusters.

According to complexity calculation, the time will increase quickly with the rapid growth of the n when n greater than k and t. Hence, the efficiency of the algorithm is decrease. So, the PAM algorithm only was used for small dataset. This is the one drawback of PAM.

Therefore, the parallel algorithm is set up by using parallel processing model. The parallel model supported to speed up linearly by increasing the computing nodes. And then, medoids-based clustering methods assign initial medoids arbitrarily. The cluster quality of the algorithm mainly depends on this value. But it is difficult to choose the best of this value. This is the second weak point of PAM. So, for finding the initial medoids in PAM clustering technique is big challenge. In the state of art, this problem is solved with the help of optimization algorithm such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Bat algorithm and Bee algorithm etc.

2.2. Apache Spark

Apache-spark, cluster computing framework is an open-source. Spark is very fast because of in-memory data processing engine [7]. The purpose of this framework speeding up the works such as batch jobs, iterative jobs, interactive query and graph processing. Spark requires the cluster manager and the data storage [8][9]. For cluster manage, Spark implement standalone, Yarn, Mesos and so on. The aim of data storage, Spark set up Hadoop Distributed File System (HDFS), MapR file system, Cassandra and so on.

The resilient distributed data set (RDD) is basic data type of Spark core engine. RDD is a programming abstraction that can be split across a computing cluster. Spark combines a driver core process that splits into tasks and distributes them among many executor processes. According to the application, the executors can be scaled up and down. Spark enables ten times faster than running on the disk. The work of Apache Spark framework show in Figure 1. The cluster manager assigns tasks to workers. Results are sent back to the driver or can be saved to disk.

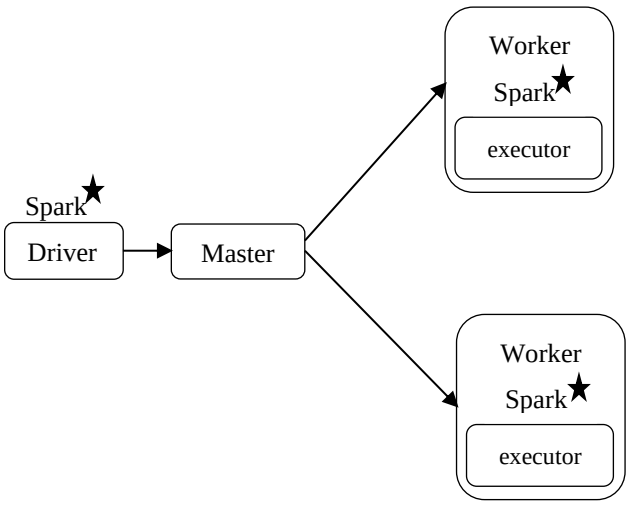

Figure 1. Apache Spark Framework

2.3. Bat Algorithm

One of the meta-heuristics, Bat algorithm works on bats behavior. The capability of the bat is search prey in complete darkness. Bats are mammals by wings and have skill of echolocation. Microbats are insectivores.

Echolocation, the active use of sonar allows microbats to avoid obstacles, detect prey, and determined roosting crevices into the dark [15].

Most bats release sound frequency to listen the echo that bounces back from the nearest objects. Bats radiate frequency differs in qualifications. The frequency is concerned with food gathering tactical. Bats produced complicated frequency that combining low frequency and high frequency. Low frequency-modulated signals to sweep during about an octave. Higher frequencies search more information such as size, range, position, speed. Signal bandwidth of bats varies depends on the species. Bat algorithm have been emphasized the characteristics of bats [15] [16].

The movement of virtual bat is given by equations (1) to (3).

$$f_i = f_{min} + (f_{max} - f_{min}).\beta \qquad f \tag{1}$$

$$v_i^t = w.v_i^{t-1} + (x_i^t - x_{best}).f_i^t \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

f is the frequency used by the bats. f_{min} and f_{max} are the minimum and maximum values. The position of the i^{th} bat denotes x_i.

v_i represent the velocity of the bat.

t indicate the current iteration, β is a random vector $\in$ [0,1].

Based on the best selected current solution, new solution is generated.

$$x_{new} = x_{old} + \varepsilon A^t \tag{4}$$

$\varepsilon \in [-1,1]$ is a random. A^t is the average loudness.

3. Proposed System

The proposed algorithm aims to obtain optimal initial medoids to get the better clusters and to reduce computation time on large volume of data. To reduce the time, the system focuses on the parallelized implementation of algorithms on Apache Spark.

The algorithms, PAM has two main drawbacks. First drawback is time complexity, and second drawback is randomly initialized. The system efforts to solve the weak point of PAM.

To solve the time complexity problem, the proposed system applied the parallel framework. Nowadays, many frameworks are popular such as Apache Hadoop, Apache Spark, Strom and so on. The PAM algorithm is iterative nature. This algorithm includes two iterative process, the distance computation and updating medoids. Apache Spark is suitable for iterative process, and very fast than other frameworks because of memory based framework. So, this proposed system implemented Apache Spark.

Figure 2 depicts the framework of parallelized algorithms on Spark. The input data is usually stored as a single file in HDFS. So, the system first need to load these input data into RDDs, where the data is horizontally split and distributed across multiple machines. In this system, Spark consists of driver process and executor process. The driver works main function and the executor actually do the assigned tasks. The proposed system configured one driver node and four worker nodes.

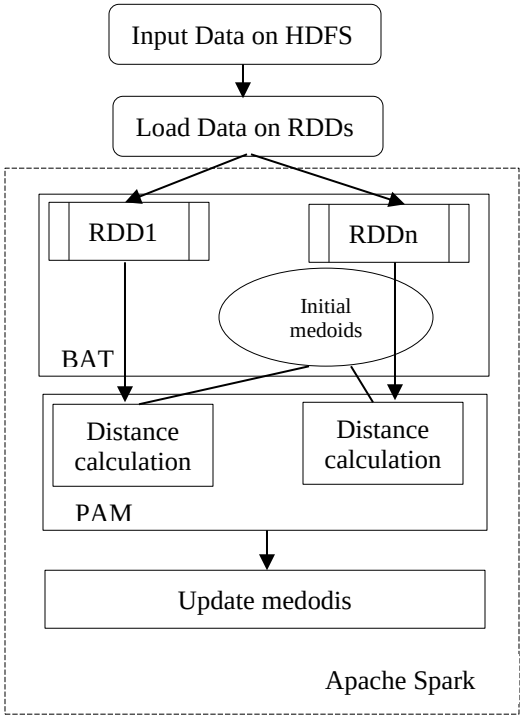

Figure 2. Parallel algorithms on Apache Spark

Another performance of clustering algorithm is cluster quality. The cluster quality is mainly depending on initial medoids. So, the proposed system aims to get the best initial medoids by combining the Bat optimization algorithm.

3.1. Hybrid PAM-Bat algorithm

The proposed algorithm implemented PAM with Bat algorithm for clustering of large application. PAM work randomized parameter to find clusters. Bat algorithm optimized the solution for better efficiency by finding the best medoids in PAM-Bat. Figure 3 shows the proposed system architecture of combining the PAM and Bat algorithm.

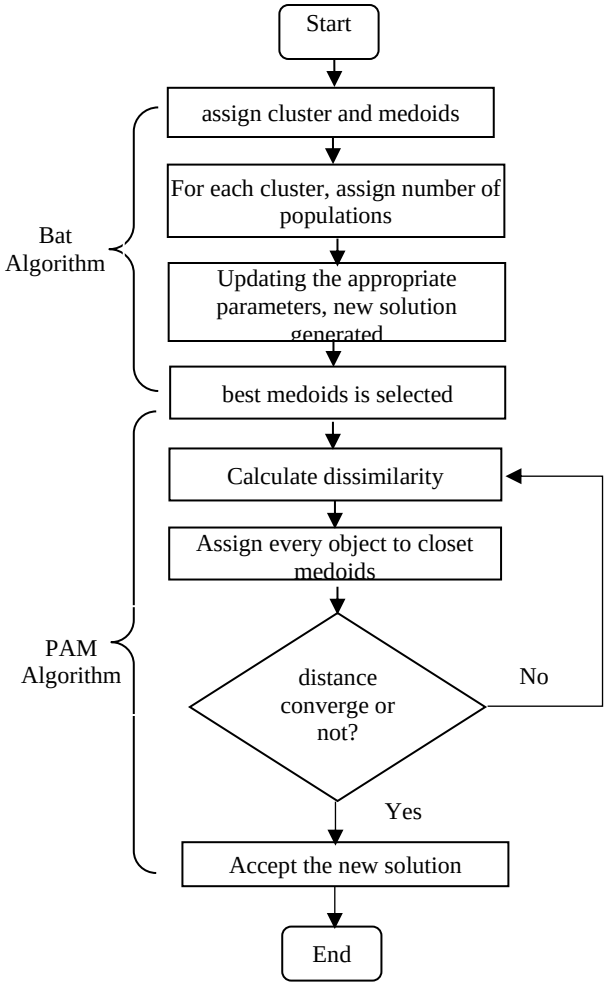

Figure 3. The proposed System architecture

The PAM algorithm achieved the best initial medoids with the help of the Bat algorithm. To implement the algorithm, randomly assigning the appropriate parameters such as velocity, position, frequency and loudness. Depend the three rules, the bat algorithms implemented. Echolocation is used to find the distance and difference. Each bat fly Velocity v_i at position x_i that varying frequency and loudness.

The best solution is selected continue until the certain criteria. The proposed algorithm assigned k-cluster to each of the N-bats. The fitness value is calculated using the equation. The data objects are assigned in cluster according to the value of fitness. Adjusting the frequency and updating the velocity, the new solution is generated. And then, the best solution is selected among the set of best solutions. Depend the accepted new solution, clusters are reassigned. The proposed algorithm as follow:

Begin
- The bat population p is initialized and the population of bat x_i where $i = 1,2,3, \dots, n$
- Define velocity v_i and set $t = 1$. Define the loudness A_i and the pulse-rate r_i and $MaxIter$. Define the frequency f_{min}. and f_{max}.

While ($t < MaxIter$) do
- Adjust frequency and update velocity and position, new solution generated.
  - a. Initialize k as initial medoids
  - b. each x_i is assign to nearest medoids object
  - c. Select a non-medoids object randomly
  - d. Calculate the swap cost
  - e. If swap cost<0
    - Change the medoids
    - Else
  - f. Go to b
- If ($rand < r_i$)
  - i. Neighbor search;
  - ii. Choose best solution;

  End if
- Calculate the new solution
- If ($rand < A_i$) then
  - Accept the new solution;
  - Increased r_i and reduce A_i

  End if;
- Find the best;
- Increase t

End while
End.

4. Experimental Result

The experiment shows the execution time of proposed system. The proposed algorithm runs the various data set sizes by parallelization method with the help of Apache Spark Framework. The system configures one master node and 4 worker nodes.

Table 1. execution time with various dataset size

Dataset Size	Time(s)	Population size
138MB	87	50
480 MB	88	50
961 MB	90	50
1.88 GB	99	50

Table 1 and Figure 4 shows time comparison among various dataset sizes. In this section, the system targets the execution time of Hybrid algorithm with various datasets size but bat population size remains unchanged.

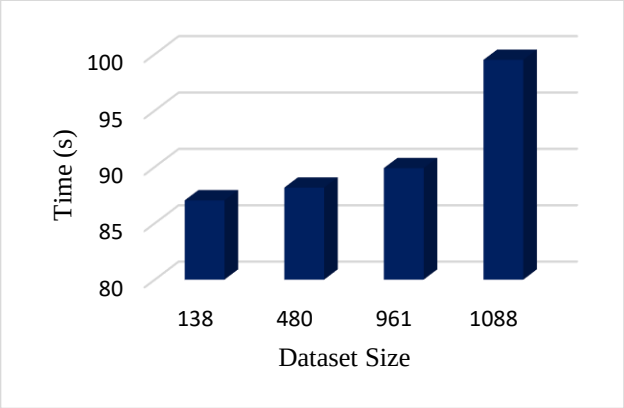

Figure 4. Comparison of time with same population sizes

According to experimental result, the hybrid algorithm handles the huge amount of dataset size on applying Apache Spark. And then, examine the proposed system and the traditional clustering method with various dataset sizes in Figure 5. For small dataset size, the traditional method slightly better than the proposed system. But, the performance of traditional method degrades for the larger dataset. Higher performance is gained the proposed system.

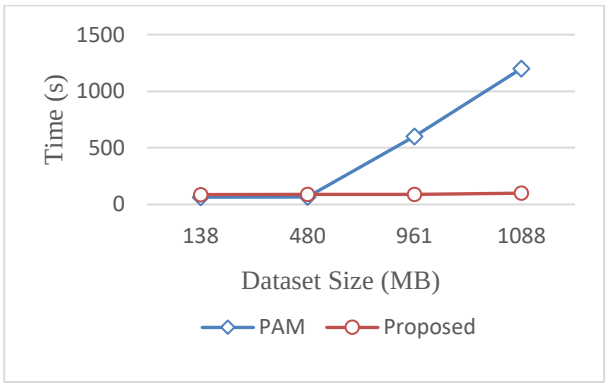

Figure 5. Execution time comparison with traditional PAM

Table 2 show the various population size apply on same dataset. The experimental result show that the higher population of bats cause higher execution time.

Table 2. Comparison of execution time with various population size

Dataset(MB)	Population Size	Time(s)
480	50	88
480	100	130
480	150	210

5. Conclusion and Future work

This paper aims to achieve the faster execution time on large volume of data clustering by using Apache Spark Framework. The experimental result of the system focuses on execution time problem. The result showed that the proposed algorithm reduced computation time on the large volume of data. In order to solve the problem of cluster quality, this system proposed hybrid algorithm which tends to find the optimal initial medoids of clustering technique. Further development of the system will measure the cluster quality of the proposed system based on various population sizes.

6. References

[1] H.S. Park, C.H. Jun, "A simple and fast algorithm for K-medoids clustering", *Expert systems with applications*, 2009 Mar 1, pp. 3336-41.

[2] Y. Jiang, J. Zhang, "Parallel K-Medoids clustering algorithm based on Hadoop", *IEEE 5th International Conference on Software Engineering and Service Science*, 2014 Jun 27, pp. 649-652.

[3] M.O. Shafiq, E. Torunski, "A parallel K-medoids algorithm for clustering based on MapReduce". *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016 Dec 18, pp. 502-507.

[4] T. Inkaya, S. Kayalıgil, N.E. Özdemirel, "Swarm intelligence-based clustering algorithms: A survey. InUnsupervised learning algorithms", *Springer*, 2016, pp. 303-341.

[5] H. Zhou, K.A. Luo, "K-medoids clustering algorithm based on Particle Swarm Algorithm with Simulated Annealing", *In Applied Mechanics and Materials*, 2013, pp. 1628-1631.

[6] R. Tiwari, M. Husain, S. Gupta, A. Srivastava, "Improving ant colony optimization algorithm for data clustering", *In Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, 2010 Feb 26, pp. 529-534.

[7] da Silva Morais T, "Survey on frameworks for distributed computing: Hadoop, Spark and storm", *In Proceedings of the 10th Doctoral Symposium in Informatics Engineering-DSIE*, 2015 Jan 29.

[8] https://searchdatamanagement.techtarget.com/ definition/ Apache-Spark

[9] https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-overview.html

[10] A. Bhat, "K-medoids clustering using partitioning around medoids for performing face recognition", *International*

Journal of Soft Computing, Mathematics and Control, 2014 Aug, pp. 1-2.

[11] X. Cui, P. Zhu, X. Yang, K. Li, C. Ji, "Optimized big data K-means clustering using MapReduce", *The Journal of Supercomputing*, 2014 Dec 1, pp. 1249-59.

[12] Y. Kim, K. Shim, MS. Kim, JS. Lee, "DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce", *Information Systems*, 2014 Jun 1, pp. 15-35.

[13] Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, J. Fan, "Mr-dbscan: an efficient parallel density-based clustering algorithm using Map reduce", *IEEE 17th International Conference on Parallel and Distributed Systems*, 2011 Dec 7, pp. 473-480.

[14] XS. Yang, "Bat algorithm: literature review and applications", *arXiv preprint arXiv*, 2013 Aug 18.

[15] XS. Yang, "A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization", *Springer*, 2010, pp. 65-74.

[16] CW. Tsai, WC. Huang, MC. Chiang, "Recent development of metaheuristics for clustering", *In Mobile, Ubiquitous, and Intelligent Computing*, 2014, pp. 629-636.

[17] T. Shohdohji, F. Yano, Y. Toyoda, "A new algorithm based on metaheuristics for data clustering", *Journal of Zhejiang University-SCIENCE A*, 2010 Dec 1.

[18] Aung NY, Mon AC, Hlaing SZ, "Performance Analysis of Parallel Clustering on Spark Computing Platform", *The 2nd International Conference on Advanced Information Technologies*, 2018.