Performance Analysis of Parallel Clustering on Spark Computing Platform

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

Aye Chan Mon
University of Information Technology
ayechanmon@uit.edu.mm

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

Abstract

In the area of information and technology, data is generated from a plethora of sources such as social media, internet of things, multimedia, sensor networks. Clustering is an essential data mining tool for analyzing this valuable information.Clustering algorithms are generally classified as a hierarchical and partitioning algorithm. This paper interested in partitioning algorithms. There are two kinds of partitioning algorithm, mean-based and medoids-based. The paper focuses on medoids-based because of medoids less influence by outliers or other extreme values than mean. But, one of the main issues of partitioning algorithm cannot handle large volume of data in case of the poor cluster quality and higher execution time.The objective of theresearchis to solve these two issues.To improve clustering quality, this paper appliesswarm intelligence optimization algorithm on the partition clustering algorithm. And then, this paper expects to reduce execution time for clustering large volume of data by using Spark framework.

Keywords- Clustering, Partitioning algorithm, Bat algorithm, Apache Spark

1. Introduction

Data mining is defined as a process used to extract usable data from a larger set of any raw data. Data mining has applications in multiple fields, like science and research. As an application of data mining, businesses can learn more about their customers and develop more effective strategies related to various business functions [15]. Data mining involves effective data collection and warehousing as well as computer processing. Data mining is also known as Knowledge Discovery in Data (KDD). Clustering means grouping the objects based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group will be similar(or related) to one another and different from (or unrelated to) the objects in other groups [16].

It is the main task of data mining and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, Bioinformatics, data compression, and computer graphics. Popular notions of clusters include groups with small distances between cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem.

Clustering algorithms are generally classified as hierarchical and partitioning algorithms. This paper is interested in partitioning algorithms. K clusters found by a partitioning method are of higher quality than the K clusters produced by hierarchical method [13]. The main objective of the partition clustering algorithm is to divide the data point into K partitions. Each partition will reflect one cluster. The technique of partition depends upon certain objective function. There are mainly four types of partitioning algorithm includes as K-Mean Algorithm, K-Medoid Algorithm (i.e. Partition Around Medoid-PAM), CLARA and CLARANS.

In K-Mean clustering algorithm, a cluster is represented by its centroid, which is usually the mean of points within a cluster. Other algorithms are medoids based algorithm. Medoids algorithms select k-medoids initially and then swap the medoids object with non-medoids thereby improving the quality of cluster. Medoids based clustering is more robust than mean-based [14]. Because, medoids less influence by outliers or other extreme values than mean. So, these paper pursuit of medoids-based clustering algorithms. Among them, PAM algorithm is less sensitive to outliers than other partitioning algorithms and easy to implement. Also, traditional medoids-based clustering algorithms cannot handle large amounts of data. The weakness of these algorithms is the large volumes of data,the less effectiveness and efficiency [1]. So, this paper is proposed with the new PAM algorithm for solving these issues. PAM will combine swarm optimization algorithm for improving the cluster quality. For solving the execution time problem, this system will be worked on Spark parallel computing platform.

The paper is structured as follows: Section 2 presents related works. Section 3 describes the theory background. Section 4 discussesthe proposed system and Section 5 provides the results of the experiment.

2. Related Works

In the recent years, many clustering algorithms have been proposed. [1] proposed a new algorithm for k-medoids clustering and test several methods for selecting initial medoids. The proposed algorithm calculates the distance matrix once and uses it for finding new medoids at every iterative step. And then, compare with the results of another algorithm. That proposed method has better performance than k-means clustering and significantly reduced computation time. Partition Around Medoid has been used for clustering of face data [2]. The results are increased robustness to noise and outliers in comparison to other clustering methods.

[3] proposed optimized big data k-mean using MapReduce in which they claimed to counter the iteration dependence of MapReduce jobs. They used a sequence of three MapReduce jobs for the purpose. In their approach sampling technique is used in the first MR job and in the final MR job the data set is mapped to centroids using the Voronoi diagram. Other clustering algorithms like density based have also been implemented in distributed platforms. Work done in [4][5] implemented the DBSCAN algorithm in MapReduce. [6] modified k-means to deal with large scale heterogeneous data sets. To the best of our knowledge of all the versions of k-means presented so far in MR, number of clusters need to be specified before the start of the algorithm. Also, for getting the optimal number of clusters multiple runs might be required.

Yang [9] developed Bat algorithm in 2010. It provided a timely review of the bat algorithm and its new variants. It reviewed and summarized wide range of diverse applications and case studies. And then, further researchtopics are discussed. Xin-She Yang [10] proposes a new Meta heuristic method. To combine the advantages of an existing algorithm into the new bat algorithm. After a detailed formulation and explanation of its implementation, they compare the proposed algorithm with other existing algorithms, including genetic algorithms and particle swarm optimization (PSO).

3. Theory Background

3.1. Partition Around Medoid (PAM)

Partition Around Mediods (PAM) is developed by Kaufman and Rousseuw in 1987. It is based on the classical partitioning process of clustering. The algorithm selects k-medoids initially and then swaps the medoids object with non-medoids, thereby improving the quality of the cluster. This method is comparatively robust than K-Mean particularly in the context of noise or outlier. Medoids can be defined as that object of a cluster, instead of taking the mean value of the object in a cluster according to reference point. K-Medoids can find the most centrally located point in the given dataset. PAM algorithm is described in the following:

PAM Algorithm:

Input:

- K:The number of clusters
- D: A data set containing n objects

Output:

- A set of K clusters

Method:

1. The algorithm begins with arbitrary selection of the K objects as medoids points out of n data points(n>K).
2. After selection of the K-medoids points, associate each data object in the given data set to most similar medoids.
3. Randomly select non-medoids object O.
4. Compute total cost S of swapping initial medoids object O.
5. If S>0, swap initial medoids with the new one.
6. Repeat steps until there is no change in the medoids.

In the PAM algorithm, the initial medoids is chosen by random. The result of such algorithm is highly depending on the initial choice of medoids and in the process of optimizing the objective function it may get stuck in local optima. To overcome these limitations, the nature inspired techniques have been proposed. These nature inspired techniques are decentralized and self-organized in behavior.The combination of PAM clustering and the nature inspired techniques makes as a hybrid approach which is very useful in the data mining.

3.2. Swarm intelligence algorithms

Swarm intelligence (SI) is one of the categories of nature-inspired problem-solving techniques. Swarm uses their environment and resources more efficiently by collective intelligence. There are many nature inspired techniques like artificial bee colony algorithm (ABC), particle swarm Optimization (PSO), ant colony optimization (ACO), bat algorithm (BA), firefly algorithm and glowworm swarm optimization (GSO). Among them, the proposed system chooses the bat algorithm because ofthis algorithm possesses the advantage of simplicity and additionally flexibility.

3.2.1. Bat algorithm. Bat algorithm is one of swarm intelligence-based algorithm which is worked on the echolocation of bats. This algorithm was developed by Xin-She Yang in 2010. Bats algorithm is based on the echolocation behavior of bats. They can find their prey or food and also, they can know the different type of insects even in a complete darkness. These bats use a type of sonar namely as echolocation. They emit a loud sound

pulse and detect an echo that is coming back from their surrounding objects.

Their pulse variation in properties and will be depend on the species. Their loudness is also varied. When they are searching for their prey, their loudness is loudest if they are far away from the prey and they will become slow when they are nearer to the prey. For the emission and detection of echo which are generated by them, they use time delay. And this time delay is between their two ears and the loudness variation of echoes. The propose system based on the echolocation behavior of bats to know the initial value to overcome the partition around medoids issue.

Bat Algorithm:

Step 1 : Set defined appropriate constant parameters

Step 2 :Bat movement is calculated by the three equations

(i) Update the frequency

$$f_i = f_{min} + (f_{max} - f_{min})\beta$$

(ii) Update the velocity

$$v_i^{t+1} = v_i^t + (v_i^{t-1} - x_*)f_i$$

(iii) Calculate the next position

$$x_i^{t+1} = x_i^t + v_i^t$$

Step 3 : The fitness function defined to be the total dissimilarity between every object and the medoids of its cluster

The traditional bat algorithm is modified to the clustering process by randomly assigning k-clusters to each of the N bats. In the next stage, fitness of medoids in each bat is computed. The data items or objects are placed in proper cluster that is based on the fitness value of medoids in a bat. In successive generations, the new solution is generated by adjusting the frequency, updating the velocity and creating new medoids values. For each bat, the best solution is selected among a set of the best solutions from the other bats. To accept new solution, the frequencyis increased and reduced loudness is considered. Based on the newly selected solution, clusters are reassigned for medoids update assignment.

In the proposed system, we combine the two algorithms for the purpose of achieving better performance in data clustering. Also, this system intends to cluster large volume of data.Large-scale data has turned to parallel and distributed processing by providing advanced mechanisms. How to wisely use existing parallel frameworks with large-scale data becomes the biggest challenge.

Hadoop MapReduce and Spark are most popular frameworks in many open-source parallel frameworks. Spark can do it in-memory, while Hadoop MapReduce has to read from and write to a disk. As a result, the speed of processing differs significantly. Spark may be up to 100 times faster. If the tasks process data again and again – Spark defeats Hadoop MapReduce. Spark's Resilient Distributed Datasets (RDDs) enable multiple map operations in memory, while Hadoop MapReduce has to write interim results to a disk. The main factor of the proposed partitioning clustering algorithm is iterative nature. Apache Spark is more suitable for iterative algorithms. So, the proposed system chooses this framework.

3.3. Apache Spark

Apache Spark is an open source big data processing framework built around speed, ease of use, and sophisticated analytics. Spark has several advantages compared to other big data and MapReduce technologies like Hadoop and Storm. Spark supports machine learning, SQL queries, graph data processing and streaming data for analysis. Spark supports languages such as Java and Python, and it is implemented in Scala, and it runs on the Java Virtual Machine (JVM).

Spark consists of cluster manager, driver program (spark context), executor or worker and HDFS. In spark, a Driver program is considered as the main program. Spark Context is for the coordination of the applications which run on clusters as a set of processes. The processes used for applications are assigned uniquely, i.e. they all have their processes and due to this task run in multiple threads, and they must connect to worker nodes. These worker nodes run computations and store the data. Programming is written in Java or Python language which is sent to the executor and it runs the tasks. The two main key concepts used in Apache Spark are Resilient Distributed Datasets (RDD) and Directed Acyclic Graph (DAG).

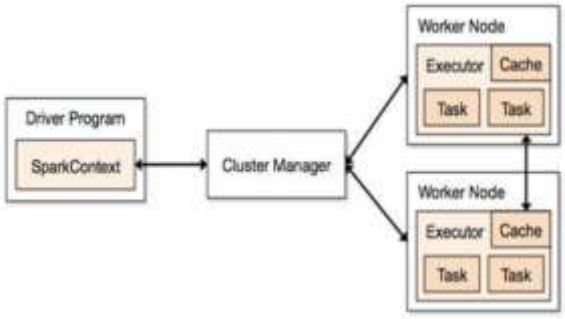


Figure 1: Apache Spark

3.3.1. Resilient Distributed Dataset (RDD). Resilient Distributed Datasets work as a collection of elements which are operated in parallel. In the distributed file system Spark runs on Hadoop cluster, and RDD is created from files in the format of text or sequence files. RDD is used for reading the objects in the collection and when some partition is lost, it can be rebuilt because RDDs are distributed across a set of machines.

4. Proposed System

The research aim is to improve the performance efficiency and effectiveness of the traditionalpartitioning clustering algorithm for handling the large amount of data. This proposed system has two portions. First, the system mentions the execution time of clustering. And then,the system discusses the cluster quality.This paper only focused on execution time.

4.1. Pre-processing

The first portiondiscussed on execution time of two algorithms. This step tested on parallel PAM and Bat algorithms by using Apache Spark.

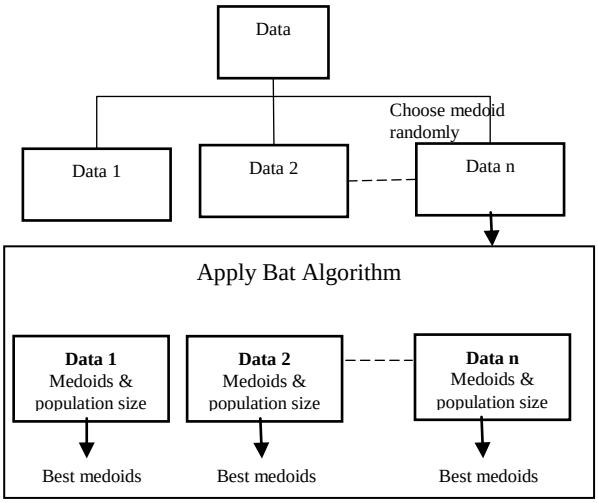


Figure 2: Parallel Bat algorithm

Figure 2 show the parallel bat algorithm. The data is distributed on the number of workers of apache spark. And then, we apply the bat algorithm for choosing the best medoids. PAM algorithm also work parallel on applying the apache spark as shown in the figure 3.

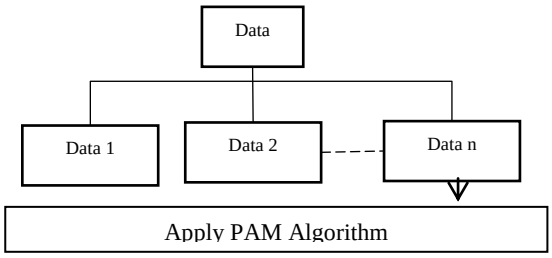


Figure 3: Parallel PAM algorithm

4.2. Bat-PAM Hybrid Algorithm

The secondpart of research work concerns about the quality of cluster results**.** Traditional PAM chooses medoids randomly. This is mainly affected of cluster quality. Bat algorithm is choosing the best of medoids by assigning the population of bats around the medoids. Best medoids that have chosen by Bat is parameters in PAM algorithm. The proposed system architecture shows in figure 4. But, this paper doesn't mention this portion.

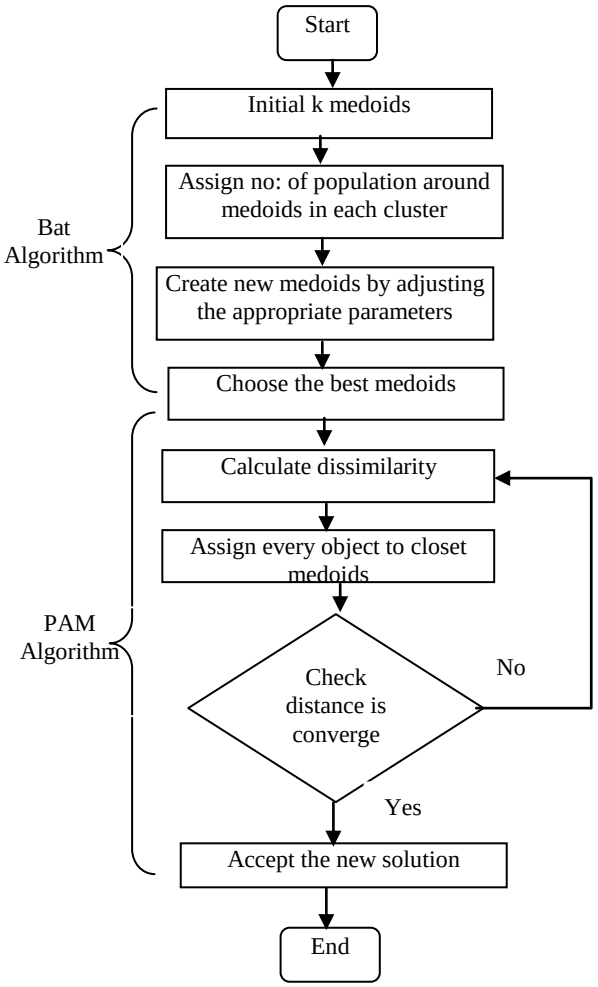


Figure 4: The Proposed System architecture

5. Experimental Result

The system tests the performance efficiency of parallelization technique of two algorithms. The system implemented on a personal computer with an Intel (R) Core (TM) i7-4770 CPU (3.40GHz) with8GB RAM.The system uses the sample Census dataset about 500MB. This dataset contains 1,000,000 records and 68 attributes.

First, we test the execution time of the dataset in sequentially. Next, we implemented the standalone

Apache Spark. Spark Standalone deployment means Spark occupies the place on top of HDFS (Hadoop DistributedFile System) and space is allocated for HDFS, explicitly.

Figure 5 shows the execution time of traditional PAM and PAM on Apache Spark. The input data is stored in Hadoop DistributedFile System (HDFS). So, we first need to load these inputdata into RDDs, where the data is split anddistributed across all nodes.The results show that parallel technique with spark framework significantly outperforms.

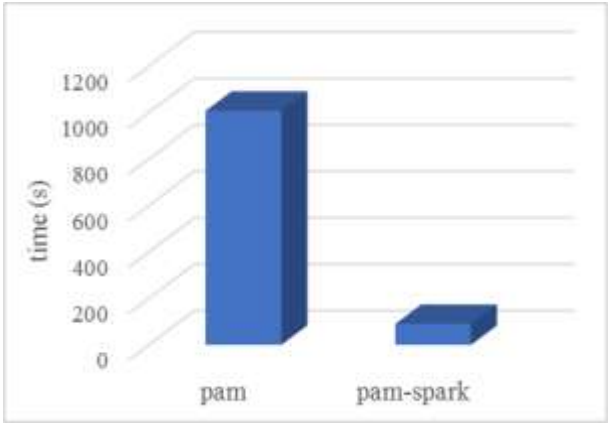
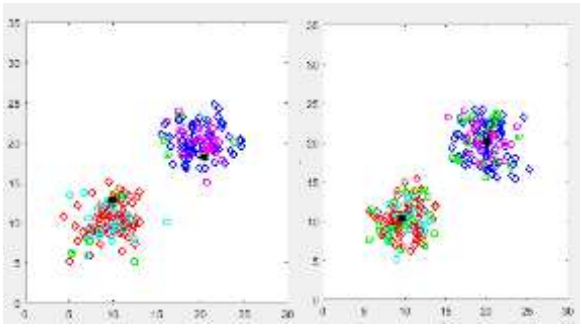


Figure 5: Comparison of execution time

The performance of apache spark is depend on the executor memory and use of cores, and the number of worker nodes. So, we consideredthese facts. At that point, the size of dataset is a factor that we need to consider. In the system, we configure one master and two worker nodes. Thiscan handle the current dataset size. In tend to use larger size of the data we will adjust this.

And then, we discussed the bat algorithm for clustering sequentially. This system simulated the number of bat population from n=5 to n=20 as shown in figure 6.



(a) n=5 (b) n=20

Figure 6: Different numbers of populations (n) on dataset

We observe the execution time of different number of populations on the dataset.

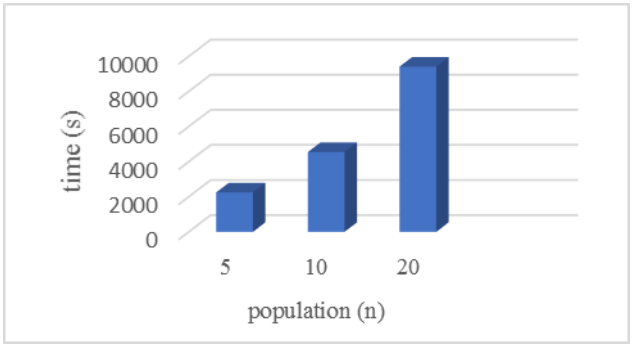


Figure 7: Comparison the runtime on the different number of bat populations

The experimental results show that the lower population size of bats causes quickly converge. Higher population size may cause slow convergence.

6. Conclusion and Future Works

Clustering is always confronted with questions such as an unstable clustering result, low executive efficiency. For solving the higher execution time problem, the system used the Apache Spark Parallel Computing Platform. Further development of this system,the Partition around medoids algorithm will combineBat algorithm to get the better cluster quality. Also, we will test the real large volume of the Myanmar census dataset. And then, we will compare the cluster quality and execution time with other well-known clustering algorithms.

7. References

[1] Hae-Sang Park, Chi-Hyuck Jun, “A simple and fast algorithm for K-medoids clustering”, Expert System with applications, ELSEVIER, 2009.

[2] Aruna Bhat, “K-medoids clustering using Partitioning arund medoids for performing face recognition”, International Journal of Soft Computing, Mathematics and Control (IJSCMC), Vol.3, No.3, August 2014.

[3] Cui, Xiaoli and Zhu, Pingfei and Yang, Xin and Li, Keqiu and Ji, Changqing, “Optimized big data K-means clustering using MapReduce”, The Journal of Supercomputing, 2014.

[4] Kim, Younghoon, Kyuseok Shim, Min-Soeng Kim, and June Sup Lee, “DBCURE-MR: an efficient density-

based clustering algorithm for large data using MapReduce", Information Systems 42, 2014.

[5] He, Yaobin, te al, "Mr-dbscan: an efficient parallel density-based clustering algorithm using mapreduce", Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on IEEE (2011).

[6] Cai, X., Nie, F., and Huang, H.., "Multi-view k-means clustering on big data", In proceeding of the Twenty-Third international joint conference on Artificial Intelligence, AAAI Press (2013).

[7] Ankita Sinha, Prasanta K.Jana, "A Novel K-Means based Clustering Algorithm for Big Data", International Conference on Advances in Computing, Communications and Informatics, 2016.

[8] Yasmine Aboubi, Habiba Drias, Nadjet Kamel, "BAT-inspired algorithm for Clustering LARge Applications", ELSEVIER (2016).

[9] Xin-She Yang, "Bat Algorithm: Literature Review and Applications", International Bio-Inspired Computation, 2013.

[10] Xin-She Yang, "A New Metaheuristic Bat-Inspired Algorithm", Computational Intelligence, Springer (2010).

[11] Tsai, C.W., Huang, W.C., and Chiang, M.C. "Recent development of metaheuristics for clustering", In Mobile, Ubiquitous, and Intelligent Computing (2014).

[12] Tsutomu, S., Fumihiko, Y., and Yoshiaki, T., "A new algorithm based on metaheuristics for data clustering", SCIENCE AISSN (2010).

[13] Yasmine Aboubi, Habiba Drias, Nadjet Kamel, "BAT-inspired algorithm for Clustering Large Applications", 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016: Troyes, France, 28-30 June 2016.

[14] A.Dharmarajan, T.Velmurugan, "Efficiency of k-Means and k-Medoids Clustering Algorithms using Lung Cancer Dataset", International journal of Data Mining Techniques and Applications, 2016.

[15]https://economictimes.indiatimes.com/definition/data-mining.

[16]https://www.slideshare.net/archnaswaminathan/cdm.