# Parallel PAM Clustering Algorithm for Learning Analytics

Nway Yu Aung, Swe Zin Hlaing
*University of Information Technology, Yangon, Myanmar*
*nwayuaung@uit.edu.mm, swezin@uit.edu.mm*

## Abstract

*Learning Analytics (LA) is defined as an area of research and application and is related to academic analytics, action analytics, and predictive analytics. This paper focuses the handling huge amount of data for better analysis. The challenges facing LA are regarding the need to increase the scope of data capture so that the complexity of the learning process can be more accurately reflected in analysis. This paper focuses on handling huge amount of data for better analysis. Partition Around Medoids (PAM) algorithm is one of the partition clustering algorithms. It tackles the problem in an iterative. However, it is not widely used for large data because of its high computational complexity. Parallelization technique can solve this problem. So, this paper proposed parallel PAM algorithm which is implemented by using Spark framework. This paper showed that the partition algorithm on Spark is slightly better than execution time tradition PAM algorithm.*

**Keywords-** Clustering, PAM, Spark

## 1. Introduction

Nowadays, Information and technology are more developed. Data are generated from various sources such as social media, Internet of Things, multimedia, sensor networks etc… This huge amount of data can be used in many fields. For example, health care, bioinformatics, public administration, educations and many more. But, handling tremendous amount of data is difficult in term of storage, processing, retrieval. Application of some preprocessing techniques can make the data comprehensible to form the data analytics. Clustering, an unsupervised data mining technique which can effectively applied in such situations. The clustering was a process that divided the abstract object into same objects classes. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Clustering is used to find the shape of data set, and also in detection of anomalies. Nowadays, the clustering algorithms include partitioning methods, hierarchical methods, density-based method, model-based method and grid-based methods. Medoids algorithms are stronger than means algorithms because medoids are less affected by noisy values. Partition Around Medoids (PAM) algorithm is one of the medoids algorithms, which attempts to determine k partitions for n objects, after an initial selection of k representatives, the algorithm repeatedly tried to make a better choice of cluster representatives and analyze all of the possible pairs of objects. But, partition around medoids algorithm has several drawbacks. This algorithm works successfully for small data sets but it does not work well for huge data sets. So, this paper intends to parallel partition Around Medoids algorithm which is proposed to handle to big data.

The data size is increasing at an exponential rate which makes it difficult to be handled on single machine environment. So, the existing algorithm need to optimize to run in distributed environment. Apache Hadoop emerged as one of the most powerful, scalable, fault tolerant platforms for this purpose. Hadoop provides two major abstractions for data storage and processing. Hadoop Distributed File Systems (HDFS) is used for distributed storage and MapReduce parallel programming is used for distributed processing. However, MapReduce programs are very much sensitive to iterations, as in each round the data is written back to the file system. Multiple read and writes to the file system increases to IO cost. PAM algorithm is iterative in nature. So, MapReduce based is quite costly.

Moreover, to get the optimal number of clusters, the algorithm needs to execute multiple times. Apache Spark, a very recently developed framework, is a better alternative. Spark does in-memory execution, which is faster in comparison to multiple read and write to the disk as in case of MapReduce. Hence, the execution time is optimized in Spark. It has been experimentally proved that spark works 100 times faster than Hadoop MapReduce when data is in memory; also the speed is 10 times when data is accessed from the disk. Spark can run on Hadoop, Mesos, standalone, or in the cloud. The proposed algorithm is implemented using Spark. The organization of the remaining paper is as follows. Section 2 reviews some of the related work. Section 3 discusses the proposed work. Experiment and results are in section 4. Conclusion of the paper is in Section 5.

## 2. Related Work

In the recent years, many clustering algorithms for big data have been proposed which are based on distributed and parallel computation. Cui, Xiaoli, et al.proposed optimized big data K-Means using MapReduce in which they claimed to counter the iteration dependence of MapReduce jobs [5].

Longhui Wang [1] has focused on Parallel algorithm based on Spark Cloud Computing Platform. They discussed MAX-MIN Ant System algorithm (MMAS) is parallelized to solve Traveling Salesman Problem (TSP) based on Spark cloud computing platform that combine MMAS with Spark MapReduce to execute the path building and the pheromone operation in a distributed computer cluster.

Jia LI [2] have proposed based on the bootstrap trails and implemented as an intelligent bootstrap library (IBL) on Spark to support efficient data clustering which obtain the trade-off between clustering efficiency and result quality.

Neha Bharill [3] has discussed the design of partitional clustering algorithm and its implementation on Apache Spark. Partitional based clustering algorithm called Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSO-FCM) which is implemented on Apache Spark to handle the challenges associated with Big Data Clustering.

Tapan Sharma [4] has simultaneously run multiple K-means algorithms with different initial centroids and values of k in the same iteration of MapReduce jobs. For initialization of initial centroids, Scalable K-means++ MR jobs have implemented and also run a validation algorithm of simplified Silhouette Index for multiple clustering outputs again in the same iteration of MR jobs. And then, the behavior of the above clustering algorithms which run on big data platforms like MapReduce and Spark jobs. Spark has been chosen as it is popular for fast processing particularly where iterations are involved.

Feng Bo [6] [7] proposed a new improved partition around medoids algorithm. This algorithm builds minimum spanning tree and then splits it to get k initial clusters with the relevant cluster centers. Experimental results show that the finding initial centers are closed to the desired cluster centers, and the improved algorithm achieved the stable clustering results and higher clustering accuracy.

## 3. Proposed System

### 3.1. PAM Algorithm

Traditional Partition Around Medoids algorithm was proposed by Kaufman and Rousseeuw in 1987. The PAM algorithm is based on the search for k representative objects or medoids among the observations of the data set. After finding a set of k medoids, clusters are constructed by assigning each observation to the nearest medoids. Next, each selected medoids m and each non-medoids data point are swapped and the objective function is computed. The objective function corresponds to the sum of the dissimilarities of all objects to their nearest medoids. The SWAP step attempts to improve the quality of the clustering by exchanging selected objects (medoids) and non-selected objects. If the objective function can be reduced by interchanging a selected object with an unselected object, then the swap is carried out. This is continued until the objective function can no longer be decreased. The goal is to find k representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object.

**Algorithm PAM**
**Input:**
$O = \{O_1, O_2, .., O_n\}$
K=number of desired clusters
**Output:**
K=set of clusters
PAM algorithm:
Randomly select k medoids from O
Repeat
For each $O_j$ not a medoid do
For each medoid $O_i$ do
Compute square error function $S_{ij}$
Find i,j where $S_{ij}$ is the smallest
If $S_{ij}<0$ then
Replace medoid $O_i$ with $O_j$
Until $S_{ij}>=0$
For each $O_i \in D$ do
Assign $O_i$ to $K_h$ where dis $(O_i,O_j)$ is the smallest over all medoids;

In the above algorithm, O means objects (1,2,…,n). $S_{ij}$ mean square error function and dis $(O_i,O_j)$ is the distance between object i and object j.

### 3.2. Apache Spark

Apache Spark is a popular open-source platform for large-scale data processing that is well-suited for iterative machine learning tasks. It is a lightning-fast cluster computing technology, designed for fast computation. Spark stores dataset in memory, which makes it a 100x faster than Hadoop. Also, it is a framework well suited to machine learning algorithms.

**Components of Spark:**

**Resilient Distributed Datasets and the Spark Core:**

The Spark Core is the foundation and provides basic I/O functionalities, task dispatching and scheduling. RDDs are basically a collection of partitioned data. These are generally created by referencing datasets in storages such as Cassandra, HBase et al., or by applying transformations such as map, reduce, and filter etc. on existing RDDs.

**Spark SQL:**

Spark SQL, a component on the Core, introduces a new data abstraction called DataFrame, for providing support for structured data. It provides a language to manipulate DataFrames in Java, Python or Scala.

**Spark Streaming:**

Spark streaming rests on the Core as well and levera on top of the Core which is proven to be ten times faster than Hadoop's disk -based Apache Mahout due to the distributed memory-based Spark architecture. It implements common algorithms to simplify large scale machine learning pipelines, like logistic or linear regression, decision trees or k-means clustering.

**MLlib Machine Learning Library:**

This is a machine learning framework on top of the Core which is proven to be ten times faster than Hadoop's disk-based Apache Mahout due to the distributed memory-based Spark architecture. It implements common algorithms to simplify large scale machine learning pipelines, like logistic or linear regression, decision trees or k-means clustering.

**GraphX:**

It is a graph-processing framework on the Core, and provides an API for graph computation that can model the Pregel abstraction, providing an optimized runtime.

### 3.3. Proposed Algorithm

One major drawback of traditional PAM is not suitable for huge amount of data. The proposed system resolved this problem by parallel technique. For parallelization, this system will work on Spark Computing platform. In this proposed system, PAM algorithm work parallelization by using Spark framework. PAM algorithm firstly selects one of the representative objects as medoids for every cluster and make partitions the other objects to the nearest cluster based on the distance with the chosen representative objects. Then try repeatedly to get a better choice of cluster representatives until the process comes to constringency. PAM algorithm takes as input dissimilarity D and produces as output a set of cluster centers or medoids. These medoids identify the clusters. Initialize cluster medoids.

$$m_1, m_2, \ldots, m_n$$

Let $n$ be the number of clusters and $m$ denote any size $n$ collection of the element $x_i$. Compute the minimum distance between data point $x_i$ and medoids $m_j$.

$$D(x_i, m_j) = min_{j=1,2,\ldots,n} D(x_i, m_j) \quad (1)$$

Where i∈j Re-compute medoids $m_j$

$$X_j \in j, m'_j = x_i \text{ and } x_{i=} m'_j \quad (2)$$

$$m_j = min\left(\sum x_i \in jD(x_i, m_j)\right) \quad (3)$$

The algorithm repeat works until medoids do not change. This system used learning analytics data sets to check the quality of spark based clustering algorithm.

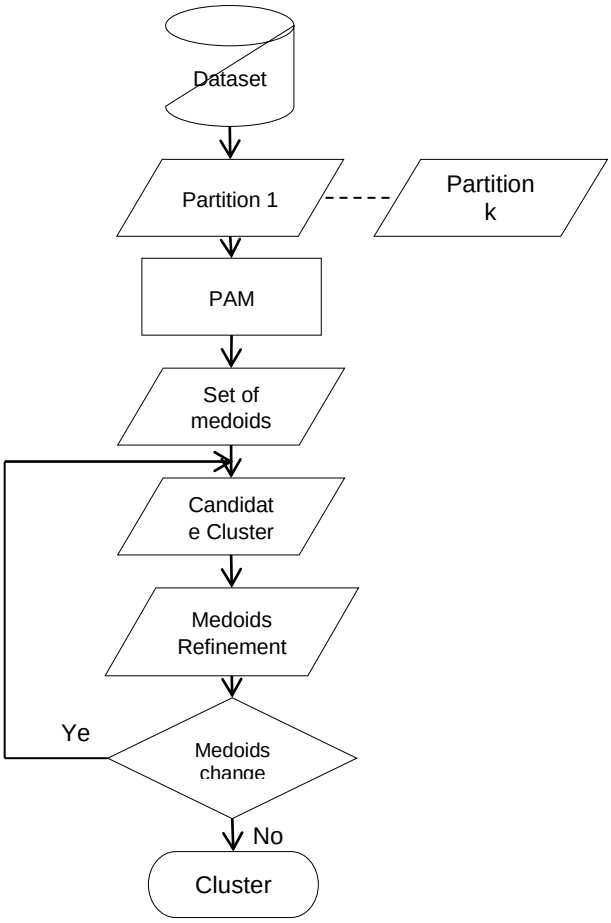


**Figure 1. System flow diagram**

*International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2012.

[7] Mark Van der Laan, Katherine Pollard, Jennifer Bryan, "A new Partitioning around medoids algorithm". *Journal of Statistical Computation and Simulation*, 2013.

[8] Isaac B.Muck, Vasil Hnatyshin, Umashanger Thayasivam, "Accuracy of class prediction using Similarity functions in PAM". *IEEE International Conference on Industrial Technology (ICIT)*, Taipei Taiwan, 2016.

[9] Abhishek Bhattacharya, Shefali Bhatnagar, "Big data and Apache Spark: A Review", *International Journal of Engineering Research and Science (IJOER)*, 2016.

[10] Jiawei Han, Micheline Kamber, Jian Pei, Data Mining, Concepts and Techniques, Third Edition, 2011.

[11] https://spark.apache.org.

## 4. Experimental Result

The proposed algorithm was simulated on UBUNTU 16.04 operating system. The Hadoop Version 2.7.0 was installed. Spark 2.0.0 run on top of Hadoop. Apache Hadoop and Spark are open source software. Figure 3 show the execution time of three algorithms: traditional PAM, PAM-Hadoop and PAM-spark.

In PAM-Hadoop, HDFS is stored the sample file and the initial center. After reading the initial cluster center, the other samples were divided to the most similar cluster parallel in Mapper function. In mapper function, (key, value) output pair is the cluster and the sample file. And then, clustering is again in Reducer function to identify the final center. In reducer function, (key, value) output pair is the cluster and the new center. The iteration is running that the result is compared with the new center which is not changed.

In PAM-spark, initial data are divided into partitions and distributed among various nodes in a cluster of computer. The data chunks are stored in Hadoop Distributed File System (HDFS). HDFS keeps 3 replicas (default) of each data chunk which minimizes the chances of data loss due to node failure. The replication factor can be increased or decreased according to requirement of the programmer. For each block in HDFS, the abstraction provided by spark, the Resilient Distributed Datasets (RDD) contains one partition. The original data is partitioned and distributed among the various nodes in the cluster, and then picked and worked upon in parallel by Spark. This system used the algebra dataset. These dataset contains 19 attributes and referenced from the website (https://pslcdatashop.web.cmu.edu). The data sets cannot actually be considered big data. However, they are large enough to judge the efficiency.

In this comparison, PAM-spark and PAM-Hadoop are faster than traditional PAM algorithm. Moreover, PAM-spark was much faster than PAM-Hadoop because of the advantages of Spark over Hadoop.

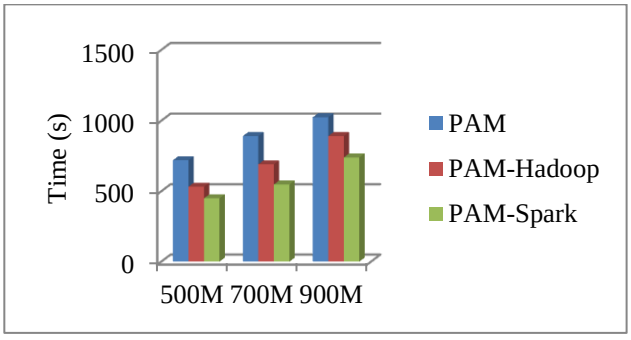


**Figure 2. Time comparison between three algorithms**

## 5. Conclusion

This paper presented partition around medoids algorithm implemented on Spark. Experimental results proved that the proposed algorithm outperforms the traditional partition algorithm implemented in Spark. Moreover the algorithm scales gracefully on increasing the data size and adds more machines to cluster. But, the proposed algorithm mainly focuses on time comparison. The proposed system should not consider the cluster quality. Initial cluster and medoids are main factor of considering cluster quality. In this algorithm, initial medoids choose randomly. The future work extends this algorithm for choosing initial medoids by using Bat algorithm. Moreover, the quality of clusters will be considered in another work. Also, the system will consider real world data analysis to improve the quality of teaching and learning.

## 6. References

[1].Longhui Wang, Yong Wang, and Yudong Xie, "Implementation of a Parallel Algorithm Based on a Spark Cloud Computing Platform", *Directory of open access Journal volume 8, issue 3*, Switzerland, 2015.

[2] Jia LI, Dongsheng LI, and Yiming ZHANG, "Efficient Distributed Data Clustering on Spark", *IEEE International Conference Cluster Computing*, Chicago, IL, USA, 2015.

[3] Neha Bharill, Aruna Tiwari, and Aayushi Malviya, "Fuzzy Based Clustering Algorithms to Handle Big Data with Implementation on Apache Spark" , *IEEE Second International Conference on Big Data Computing Service and Applications*, Oxford, UK, 2016.

[4] Tapan Sharma, Dr. Sunil Mathur, and Dr. Vinod Shokeen, "Multiple K Means++ Clustering of Stallite Image Using Hadoop MapReduce and Spark", *International Journal of Advanced Studies in Computer Science and Engineering, IJASCSE volume 5 issue 4*, 2016.

[5] 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 Volume 70, Issue 3*,2014.

[6] Feng Bo, Hao Wenning, Chen Gang, Jin Dawei, Zhao Shuining "An Improved PAM algorithm for optimizing Initial Custer Center", *IEEE 3rd*