ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Big Data Analysis Using Fuzzy Clustering Algorithms Implemented on Spark Framework

Divyashree. V[1], Deepika. N[2]

M. Tech Student, Department of Computer Science and Engineering, New Horizon College of Engineering, Outer Ring Road, Panathur Post, Kadubisanahalli, Bangalore, India

Senior Assistant Professor, Department of Computer Science and Engineering, New Horizon College of Engineering, Outer Ring Road, Panathur Post, Kadubisanahalli,, Bangalore, India

ABSTRACT: A huge amount of data containing useful information, called Big Data, is generated on a daily basis. For processing such tremendous volume of data, there is a need of Big Data frameworks such as Hadoop MapReduce, Apache Spark etc. Among these, Apache Spark performs up to 100 times faster than conventional frameworks like Hadoop Mapreduce. we focus on the design of partitional clustering algorithm and its implementation on Apache Spark. In this paper, we propose a partitional based clustering algorithm called Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) which is implemented on Apache Spark to handle the challenges associated with Big Data Clustering. Experimentation is performed on several big datasets to show the effectiveness of SRSIO-FCM in comparison with a proposed scalable version of the Literal Fuzzy c-Means (LFCM) called SLFCM implemented on Apache Spark.

KEYWORDS: Apache Spark, Big Data, SRSIO-FCM, LFCM, SLFCM.

I. INTRODUCTION

A Huge amount of data gets collected everyday due to the increasing involvement of humans in the digital space. We share, store and manage our work and lives online. For example, Facebook stores more than 30 Petabytes of data, and Walmart's databases contain more than 2.5 petabytes of data. Such huge amount of data containing useful information is called Big Data. It is becoming increasingly popular to mine such big data in order to gain insights the valuable information that can be of great use in scientific and business applications. Clustering is the promising data mining technique that is widely adopted for mining valuable information underlining unlabeled data. Over the past decades, different clustering algorithms have been developed based on various theories and applications. Among them, partitional algorithms are widely adopted due to their low computational requirements; they are more suited for clustering large datasets. We propose Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) implemented on Apache Spark to tackle the challenges associated with fuzzy clustering for handling big data. The proposed approach works by partitioning the data into various subsets and then performs clustering on each subset.

II. LITERATURE SURVEY

In the year of 2014, the authors Y. Wang, L. Chen, and J.-P. Mei. revealed a paper titled "Incremental fuzzy clustering with multiple medoids for large data" and describe into the paper such as a critical strategy of information investigation, grouping assumes an essential part in finding the fundamental example structure installed in unlabeled information. Grouping calculations that need to store every one of the information into the memory for examination get to be distinctly infeasible when the dataset is too vast to be put away. To handle such extensive information,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

incremental bunching methodologies are proposed. The key thought behind these methodologies is to discover delegates (centroids or medoids) to speak to every bunch in every information lump, which is a parcel of the information, and last information investigation is done in light of those recognized agents from every one of the pieces. In this paper, we propose another incremental bunching approach called incremental numerous medoids-based fluffy grouping (IMMFC) to handle complex examples that are not reduced and very much isolated. We might want to research whether IMMFC is a decent contrasting option to catching the hidden information structure all the more precisely. In the year of 2010, the authors Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst. revealed a paper titled "Haloop: efficient iterative data processing on large clusters" and describe into the paper such as the developing interest for extensive scale information mining and information investigation applications has driven both industry and the scholarly world to outline new sorts of exceedingly adaptable information escalated figuring stages. The point by point issue definition, overhauling rules determination, and the top to bottom investigation of the proposed IMMFC are given. Trial examines on a few huge datasets that incorporate genuine malware datasets have been led. IMMFC outflanks existing incremental fluffy bunching approaches as far as grouping exactness and power to the request of information. These outcomes show the colossal capability of IMMFC for huge information examination.

III. A NEW PROPOSAL FOR FEATURE SELECTION

We propose Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) implemented on Apache Spark to tackle the challenges associated with fuzzy clustering for handling big data. The proposed approach works by partitioning the data into various subsets and then perform clustering on each subset. Apache Spark performs up to 100 times faster than conventional frameworks like Hadoop Mapreduce. MapReduce's processing style can be just fine if your data operations and reporting requirements are mostly static and you can wait for batch-mode processing. But if you need to do analytics on streaming data, like from sensors on a factory floor, or have applications that require multiple operations, you probably want to go with Spark. Most machine-learning algorithms, for example, require multiple operations. Common applications for Spark include real-time marketing campaigns, online product recommendations, cyber security analytics and machine log monitoring.

IV. APACHE SPARK

Apache spark is originally developed at UC Berkeley in 2009, it is a powerful open source processing engine built for sophisticated data analysis. It has quickly become one of the most well recognized tools, giving tough competition to Hadoop MapReduce. It was adopted by IT giants such as Yahoo, Baidu and Tencent eagerly deployed Spark on a massive scale. They collectively process petabytes of data on their Spark clusters.MapReduce's processing style can be just fine if your data operations and reporting requirements are mostly static and you can wait for batch-mode processing. But if you need to do analytics on streaming data, like from sensors on a factory floor, or have applications that require multiple operations, you probably want to go with Spark. Most machine-learning algorithms, for example, require multiple operations. Common applications for Spark include real-time marketing campaigns, online product recommendations, cyber security analytics and machine log monitoring. It require numerous operations, you most likely need to run with Spark. Most machine-learning calculations, for instance, require various operations. Basic applications for Spark incorporate constant showcasing effort, online item proposals, digital security examination and machine log observing. Spark cluster consists of many machines, each of which is referred to as a node. There are two main components of Spark: Master and Workers. There is only one master node and it assigns jobs to the slave nodes. However, there can be any number of slave nodes. Data can be stored in HDFS or in local machine.

V. EXISTING SYSTEM

The literal Fuzzy c-Means with alternating optimization (LFCM/AO) algorithm performs clustering on entire dataset but it does not work well for big data. But there are many sampling methods which compute cluster centers on sampled data which is randomly selected from a huge dataset. Some of the popular sampling based methods are CLARA , CURE and the coresets algorithms. These algorithms work well for crisp partitions. But they suffer from overlapping cluster centers if the sampled data is not representative of the entire data.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Disadvantages

1. suffer from overlapping cluster centers

VI. PROPOSED SYSTEM

We propose Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSIO-FCM) implemented on Apache Spark to tackle the challenges associated with fuzzy clustering for handling big data. The proposed approach works by partitioning the data into various subsets and then performs clustering on each subset.

Advantages

1. Apache Spark performs up to 100 times faster than conventional frameworks like Hadoop Mapreduce.

VII. SLFCM ANALYSIS

The SLFCM algorithm is implemented on Apache Spark. In SLFCM, we perform the calculation of cluster membership degree in parallel on various worker nodes thus reducing the run-time as compared to linear execution on a single machine. The membership degrees are then aggregated on master node and cluster centre values are computed. This process is repeated until no significant difference is observed in the values of cluster centres. The workflow of SLFCM and the internal working of SLFCM is done on Apache Spark.

A. SRSIO-FCM

It partitions the data into equal sized subsets where in each subset the data points are chosen at random from big data without replacement. Initial cluster centers for the first subset are chosen randomly. SRSIO-FCM, like RSIO-FCM, computes cluster centers (V 1) and membership information (I 1) for the first subset and feeds V 1 as input to the second subset. It then finds the cluster centers (V 2) and membership information (I 2) for the second subset. The motivation for using the final cluster centers of one subset to initialize the cluster centers for the next subset comes from the observation that the cluster centers for the two subsets are nearer to each other. Thus, the algorithm will converge to optimal cluster centers faster as compared to the case when the cluster centers are initialized randomly.

VIII. SYSTEM ARCHITECTURE

We assess the time and space multifaceted nature of each of the proposed VL variations of FCM/AO. All operations and storage room are considered unit costs. We don't accept economies that may be acknowledged by uncommon programming traps or properties of the conditions included. For instance, we don't make utilization of the way that the portion networks are symmetric frameworks to decrease different checks from n 2 to n(n − 1)/2, and we don't expect space economies that may be acknowledged by overwriting of clusters, and so forth. Along these lines, our "correct" evaluations of time and space intricacy are correct just with the presumptions we have used to make them. Imperatively, be that as it may, the asymptotic evaluations for the development in time and space with n, which is the quantity of items in X, are unaffected by changes in tallying techniques. We show Resilient Distributed Datasets [RDDs], a circulated memory reflection that gives developers a chance to perform in-memory calculations on extensive groups in afault-tolerant way. RDDs are propelled by two sorts of uses that present figuring structures handle wastefully: iterative calculations and intelligent information mining apparatuses. In both cases, keeping information in memory can enhance execution by a request of extent. To accomplish adaptation to non-critical failure productively, RDDs give a confined type of shared memory, in view of coarse grained changes as opposed to fine-grained upgrades to shared state. Nonetheless, we demonstrate that RDDs are sufficiently expressive to catch a wide class of calculations, including late specific programming models for iterative

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

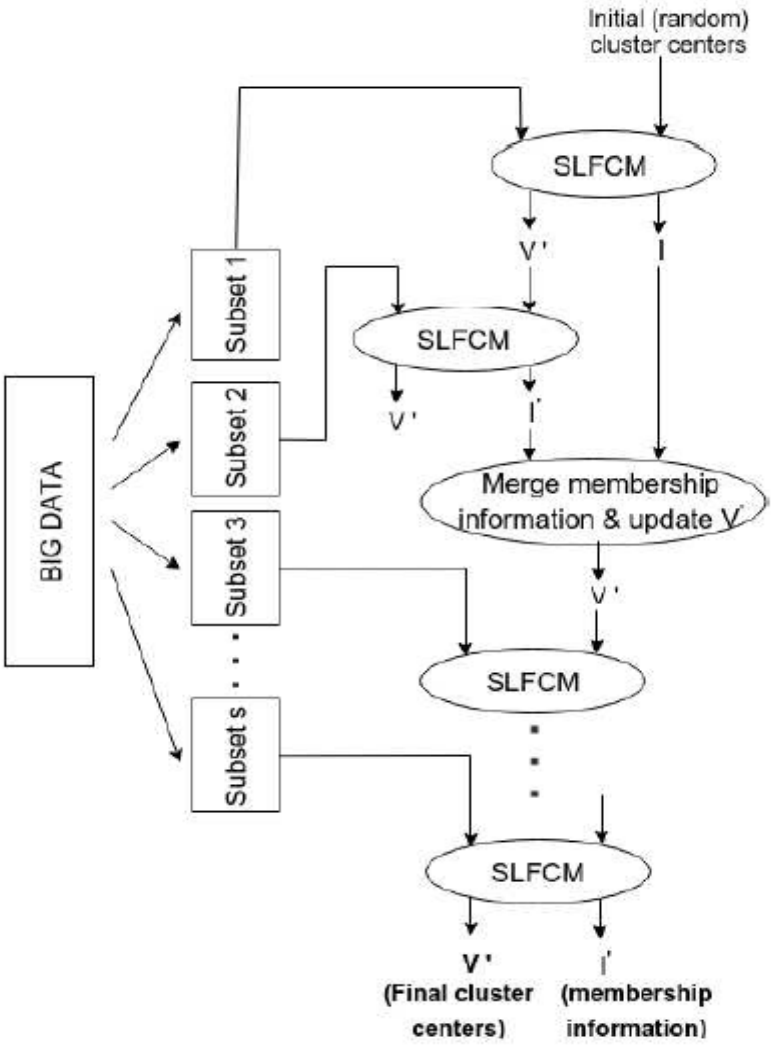

Fig 1. System Design

XI. EXPERIMENT DESCRIPTION

For the tests we selected fifteen data sets Arrhythmia, Cylinder-band, Hypothyroid, Kr-vs-Kp,Letter, Mushroom, Nursery, [7]OptiDigits, Pageblock, Segment, Sick, Spambase and Waveform5000. All of these data sets have their own properties like the domain of the data set, the kind of attributes it contains, and tree size after training.We tested each data set with four different classification tree algorithms: J48, REPTree, RandomTree and Logistical Model Trees. For each algorithm both the test options percentage split and cross-validation were used. With percentage split, the data set is divided in a training part and a test part. For the training set 66% of the instances in the data set is used and for the test set the remaining part. Cross-validation is especially used when the amount of data is limited.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

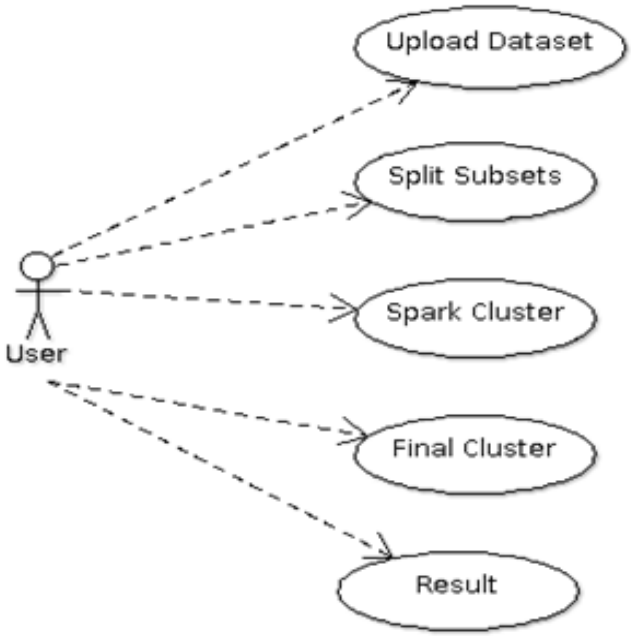

Fig 2. Use Case Module

X. SIMULATION RESULTS

The experiments were run on a Spark cluster with nodes. Each node has the following configuration: Intel Core I3(r) CPU ES-1607 v3 @ 3.10GHz x 4, 32GB RAM and 2TB storage. The algorithms were implemented in Java and tested over Hadoop version 2.4 with Apache Spark version 1.4.0. We used HDFS for storing data across the cluster and YARN for resource management. We compare the performance of SRSIO-FCM, SLFCM and SrseFCM algorithm on data sets. We implemented SRSIO-FCM on both Hadoop Mapreduce and Spark to show that Spark framework is faster than Hadoop in processing Bigdata.

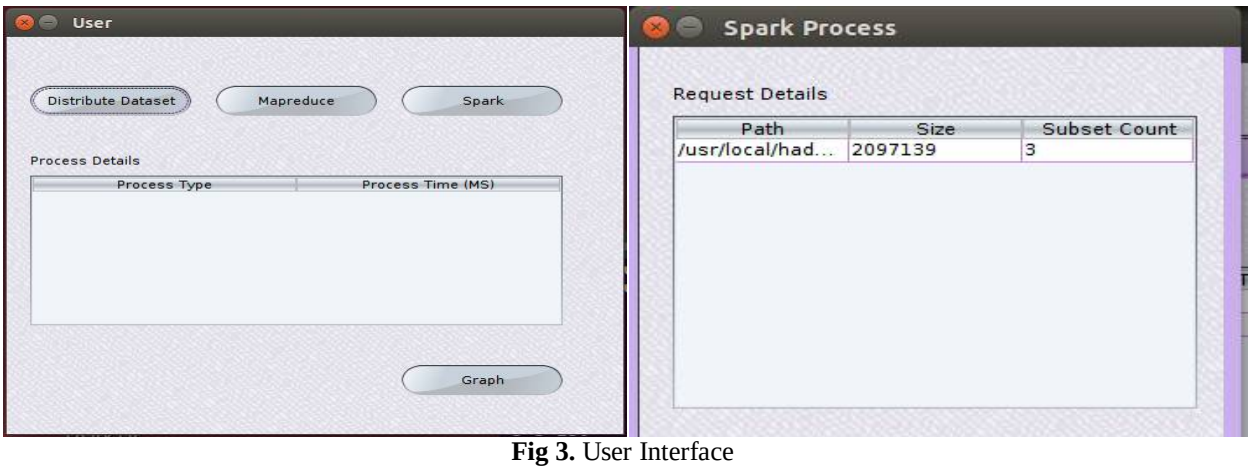

Fig 3. User Interface

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Fig 4. Performance comparison between Hadoop Mapreduce and Spark

XI. CONCLUSION

We have projected an additional Scalable Random Sampling with Iterative Optimization Fuzzy c-Means approach called SRSIO-FCM for Big Data examination. SRSIO-FCM forms Big Data piece by lump. One particular normal for SRSIO-FCM is that it takes out the issue of sudden increment in the quantity of cycles that happen amid the grouping of any subset because of the sustaining of profoundly veered off bunch focuses, created from the past subset, as a contribution for the bunching of current subset. We implemented SRSIO-FCM on both Hadoop Mapreduce and Spark to show that Spark framework is faster than Hadoop in processing Bigdata.

REFERENCES

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," pp. 1–137, 2011.
[2] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," Pattern recognition, vol. 41, no. 1, pp. 176–190, 2008.
[3] A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern recognition letters, vol. 31, no. 8, pp. 651–666, 2010.
[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," ACM computing surveys (CSUR), vol. 31, no. 3, pp. 264–323, 1999.
[5] R. O. Duda, P. E. Hart et al., Pattern classification and scene analysis. Wiley New York, 1973, vol. 3.
[6] M. Steinbach, G. Karypis, V. Kumar et al., "A comparison of document clustering techniques," in KDD workshop on text mining, vol. 400, no. 1. Boston, 2000, pp. 525–526.
[7] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 2013.
[8] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-means algorithms for very large data," IEEE Transactions on Fuzzy Systems, vol. 20, no. 6, pp. 1130–1146, 2012.
[9] P. Hore, L. O. Hall, and D. B. Goldgof, "Single pass fuzzy c means," in Proc. IEEE International Conference on Fuzzy Systems (FUZZIEEE). 2007, pp. 1–7.
[10] P. Hore, L. O. Hall, D. B. Goldgof, Y. Gu, A. A. Maudsley, and A. Darkazanli, "A scalable framework for segmenting magnetic resonance images," Journal of signal processing systems, vol. 54, no. 1-3, pp. 183–203, 2009.

BIOGRAPHY

Ms. Divyashree. V[1], MTech Student, Department of Computer Science and Engineering, New Horizon College of Engineering, Outer Ring Road, Panathur Post, Kadubisanahalli,, Bangalore - 560087.

Ms. Deepika. N[2], Senior Assistant Professor, Department of Computer Science and Engineering, New Horizon College of Engineering, Outer Ring Road, Panathur Post, Kadubisanahalli,, Bangalore - 560087