A Survey Paper on Parallel Power Iteration Clustering for Big Data

Surbhi K. Solanki, Jalpa T. Patel,
Department of Information Technology,
Shri S'ad Vidya Mandal Institute of Technology

***Abstract*—In today's research, Distributed Data Mining is the most important and popular topic because data is increasing day by day in many fields such as flight booking, passenger details etc. Vast research has been conducted in distributed classification, association, clustering etc. There are some issues in distributed data mining such as homogeneous vs. heterogeneous, fragmentation of data, replication of data, communication and computation cost, Data skewness, Integration of results, etc. Mainly this paper is conducted in the context of reduction of computational as well as communication cost for data transfer. Many clustering methods have been developed and still popularly used. However, these methods generally lack robustness and suffer from "curse of dimensionality", and computationally expensive for big data which is unstructured. Moreover, other method which is graph based method is newly developed called as Power iteration clustering (PIC). In PIC, for finding largest eigenvector power iteration method is used and then after it apply k-means algorithm for clustering purpose. It is a fast, simple and scalable method for graph clustering. But for big data it is still not scalable. In this paper our aim is to study about parallel approach called Parallel Power Iteration Clustering which can handle large data.**

KeyIndex Terms—Distributed Data Mining; Hadoop; Parallel Power Iteration Clustering; Power Iteration Clustering

I. INTRODUCTION

The task of extracting useful and interesting knowledge from large data is called Data mining. When data is too large then distributed data mining is carried out. Several data mining techniques are available such as classification, clustering, Association mining etc [1]. Now a day's distributed data mining has become popular. Datasets hosted by local computers are stored in local databases.

For increasing the performance of traditional data mining systems, Distributed data mining and parallel data mining are used and it can help to speed up the data mining process; however, they are different in several ways [2]. In distributed data mining, communication takes place with message passing and nothing is shared between processors, whereas in parallel data mining memory is shared between processors [3]. Wide research has been conducted in distributed classification, association, clustering etc. Among them distributed clustering is an important aspect of distributed data mining. There are various models of clustering and algorithms by which clustering can be done. Hierarchical clustering is connectivity based and partitioning clustering which is Centroid based clustering and it is non-hierarchical. Density based clusters are defined as areas of higher density than the remainder of the data set and also used for finding the non-linear shapes structure. Partitioning the space into defined number of cells which forms a grid on which all of the operations for clustering are performed is called Grid based clustering. Clustering of vertices on bases of edge structure is called graph based clustering [4].

Traditional methods of clustering have drawbacks. Therefore, new algorithm i.e. power iteration clustering (PIC) has developed. PIC is scalable graph based clustering approach. It is good at clustering accuracy and also fast on large datasets. But it is not suitable for large data [5]. Parallel power iteration clustering overcomes the drawback of PIC. Here computational procedures are carried out in parallel so that simultaneous computations can be performed. Hence it is cost effective solution and effective clustering of large data can be done [6].

The paper is organized as follows: In section I contains introduction, section II contains related work, section III contains power Iteration clustering, section IV contains parallel power iteration clustering, section V contains Hadoop framework, and section VI contains conclusion and future work.

II. RELATED WORK

There are so many traditional methods of clustering but popularly used traditional methods are hierarchical methods and partitioned methods given in [7]. Weizhong Yan et al**.** in [6] presented that traditional methods of clustering have drawbacks of robustness and suffer from curse of dimensionality and also they are computationally expensive for big data. To address these drawbacks, new clustering algorithms viz. grid based, evolutionary clustering, normalized cut etc have been introduced [6]. Then a modern approach of clustering algorithm has been introduced called spectral clustering. It is most important clustering algorithm. This method is based on Eigen decompositions of affinity. Spectral clustering based on affinity matrix of data points extracts the Eigen vectors having largest Eigen values and then map those vectors in 3-D Space. With the data points, clustering can be done easily in conventional way is called affinity based spectral clustering [8]. In other clustering method e.g. k-means assumes that resulting clusters are always of convex sets where as spectral clustering can solve problems with different cluster shapes and thus it results in superior performance of clustering. Drawback of spectral clustering is that it takes more time i.e. O (n^3) and space i.e. O (n^2), for eigenvalue decomposition of n-by-n affinity matrix where n is the number of data points [6].

Lin and Cohen [5] have introduced power iteration clustering. It is fast and simple algorithm for big data compared to traditional one. It replaces the eigenvalue decomposition of the similarity matrix required by spectral clustering by matrix vector multiplications, which reduces computational complexities. It is scalable graph based clustering approach.

W. Kim given in [9] that almost all clustering approach requires iterative procedures to find out optimal solution. In addition to this, rarely real-life data presents unique clustering solution and also hard to interpret its cluster representation. Therefore, parallelization of clustering approach comes in picture. In master-slave architecture, they have observed that most of the parallel clustering algorithms uses message-passing model.

Du. and Lin. in [10] presents Novel parallelization approach for hierarchical clustering on a cluster of computer nodes based on Message passing interface (MPI) for communication. Main step for analysis of gene expression is the identification of groups of genes that shows similar expression patterns. For this hierarchical clustering is developed. But limitation with this method is that it can’t handle large data sets. For solving this problem, parallelization approach is used. By experimental results, they achieved reduction in computational time and internode communication, especially for large datasets. Also it is practicable approach for high dimensional gene expression.

Martin Ester et al. given in [11], for the class identification task in spatial databases clustering algorithm is required. Moreover for large spatial databases, there are limitations for minimal requirements of domain knowledge to determine input parameters, arbitrary shape cluster discovery and good efficiency. The well know clustering algorithm offer no solution for this requirement. Therefore their paper presents new DBSCAN clustering algorithm based on density based notion of clusters. Discovering of arbitrary shape clusters is carried out by DBSCAN. This algorithm requires only one input parameter and supports its users for determining an appropriate value for it. Their experimental result shows that DBSCAN is more efficient and effective in discovering arbitrary shape clusters than CLARANS.

Chen et al. in [12] showed that for large data set spectral clustering suffers from scalability problem in both memory use and computational time and their parallel algorithm works effectively for large data. PIC has slightly better capability in handling large data but still it requires data and the similarity matrix both to fit in to computer memory, and this is impossible for large datasets. So for this need of clustering big data Weizhong Yan et al. in [6] have investigated a P-PIC algorithm which has parallel implementation of Power iteration clustering.

Here, parallel power iteration clustering uses different nodes so there is more possibilities to occur node failure, so to make the system more robust and to avoid node failure D. Jayalatchumy et al. in [13] showed that using Map-Reduce of Hadoop framework we can eliminate these issues. Because Hadoop has the facility to create replicas, if any node is going to fail or crash then its replica can be used.

III. Power Iteration Clustering

In power iteration clustering (PIC), the name includes two things first is power iteration method and second is clustering. In Power Iteration Method it finds the largest eigenvalue for developing clusters. Power Iteration Clustering is simple and scalable clustering method in terms of space and time.

F. Lin, W.W. Cohen presents in [5] by using truncated power iteration on a normalized pair wise similarity matrix of the data Power Iteration clustering finds a very low dimensional embedding of a dataset. This method is running 1000 times faster than NCut implementation based on IRAM eigenvector computation technique on large datasets. In Spectral clustering, embedding is formed by the bottom eigenvector of Laplacian of similarity matrix. In Power Iteration Clustering, embedding is an approximation to an eigenvalue-weighted linear combination of all the eigenvectors of a normalized similarity matrix. Embedding method used in PIC became effective for clustering. In comparison with spectral clustering, cost of explicit calculating eigenvectors is replaced by matrix vector calculation. They demonstrated that without sampling, grouping, or other preprocessing of data by implementing Power Iteration Clustering method on a single machine within a second it is able to partition a network dataset of 100 million edges.

Mathematical notations and background of Power Iteration clustering [5]:

- Similarity between two data points [6]:

$$S\,(x_i, x_j) = \exp\left(-\frac{||xi-xj||2}{2\sigma 2}\right)$$

  - Where, σ is a scaling parameter
  - Given Dataset is X= {x_1, $x_{2\ldots}$ x_n)

  Similarity function is a function, s(x_i, x_j); Where, s(x_i, x_j)= s(x_j, x_i) and s>=0 if i ≠ j and If i=j then s=0;

- Affinity matrix :
  Affinity matrix is defined as,
  $\mathbf{A_{ij} = s (x_i, x_j)}$ **[5]**
- Diagonal matrix:
  The degree matrix associated with affinity matrix is called Diagonal matrix, D_{ii}
- Normalized Affinity matrix :
  Normalized affinity matrix is defined as,
  $\mathbf{W= D^{-1} A}$ [5] (1)
- Laplacian matrix :
  Normalized affinity matrix (W) is related to normalized Laplacian matrix and is given by,
  L = I-W (2)
  Where, I is the identity matrix
  Therefore from equation (1) and (2),

$\mathbf{L=I- D^{-1} A}$ [5]

Partition of graph W that approximately maximizes the Normalized cut criteria is defined by Laplacian matrix (L) which has the smallest eigenvector than its second smallest eigenvector. Thus, second-smallest, third-smallest, fourth-smallest…upto the kth smallest eigenvectors of Laplacian matrix (L) will create cluster of graph W into K components [5]. For finding the largest eigenvector is called Power iteration (PI) and is also called power method. It is an iterative method that starts with an arbitrary initial vector $v^0 \neq 0$. This method repeatedly updates as under:

$\mathbf{v^{t+1} = \boldsymbol{cW}v^t}$ [5]

Where, $c=1/\| Wv^t \|_1$ is a normalizing constant to keep vector v^t from getting too large and v^t and v^{t+1} are the vectors at tth iteration and t+1th iteration respectively.

Define velocity at t iteration is $\delta^{t+1} = v^{t+1} - v^t$ and acceleration at t iteration is $\varepsilon^t = \delta^t - \delta^{t-1}$

- Steps for Power Iteration Clustering [5], [6]:

Initially the data set is considered for clustering. The similarity for each data point is calculated. The similarity point is stored in an affinity matrix A. The similarity measure finds the distance of one data point to the other. The degree of the affinity matrix A is calculated by summing the row values and stored in matrix D (Diagonal matrix). The degree represents the closeness of the data points for the clustering. A Laplacian matrix is done by subtracting the degree and similarity matrix from the Identity matrix means the value data point is reversed. From the concept of eigenvectors the data points are represented by the nature of convergence. From the property that, the larger eigenvector of matrix W is smallest eigenvector of matrix L as L is derived from W. At each iteration the vector of the data point is updated. The updation is done by multiplying the largest eigenvector found with the previous vector position of data point to get the current position of the data point.

IV. Parallel Power Iteration Clustering

Power iteration clustering algorithm has little bit better capability in handling large data. But still it requires more memory. PIC is sequential algorithm and takes more time for communication. These constraints challenged us to go for parallel clustering approach that distributes the computation and data across multiple machines in parallel so that simultaneous computation can be done. Hence it is cost effective solution and effective clustering of large data can be done. W. Kim presents in [9] in master-slave architecture they have observed that most of the parallel clustering algorithms used message-passing model.

Weizhong Yan et al. in [6] have attempted to expand data scalability of PIC by implementing a parallel power iteration clustering. Here they focus on exploring different parallel approaches and implementation details so that computation and communication costs can be minimized, and also paid great attention to ensure that the algorithm works well on low-end commodity computers. The message passing interface is a message passing library interface and same is the standard for performing communications in parallel programming [6]. They choose MPI as programming approach for implementing the parallel PIC algorithm due to its efficiency and performance for data communications in distributed cluster environments. Parallel power iteration clustering solves the problem of memory issue for storing the similarity matrix by splitting the data into small chunks and distributing these small chunks of data to multiple processors [6].

- Steps for parallel Power Iteration Clustering [6], [13]:

Input is considered as large data set. Calculate the splits and assign starting indices and end indices to slave processors. Calculation of Similarity sub-matrix and Normalization of the same is carried out by slave processor. Collection of row sums from slave processors by master processors. Concatenation of all row sums and thereby obtain Overall row sum by master processor. Master processor also generates initial vector and then broadcasts the same to all slave processors. Updation of vectors of slave processors is done by matrix vector calculation method. by matrix vector calculation method. All updated vectors are collected by Master processor from slave processors. Stop till the criteria i.e. acceleration is achieved. Finally, we get the output clusters.

With P-PIC problem of node failure occurs, to recover data during failure of one node. If there is no mechanism then it would be difficult to handle large data. So, to overcome this issue of failure of node and loss in data they used Hadoop framework [14], [15]. Table I illustrates comparison of different traditional algorithms and P-PIC.

Table I: Comparison of different algorithms:

NCutE	NCutI	PIC	P-PIC
- To find all eigenvectors, it uses classic eigenvalue decomposition method -Slower than NCutI, PIC and P-PIC. -NCutE was not run on largest synthetic datasets due to memory constraint.	-Uses Implicitly Restarted Arnoldi Method (IRAM). -Faster than NCutE but Slower than PIC and P-PIC. -NCutI was not run on largest synthetic datasets due to memory constraint.	-Uses Matrix Vector multiplication -Faster than NCutE and NCutI. -As compared to P-PIC, it can’t handle large data and takes more time.	-Uses parallel approach of PIC. -It can handle large data efficiently as compared to PIC.

V. Hadoop

Hadoop is a framework that can solve distributed file systems for large datasets using a simple programming model i.e. MapReduce. It uses clusters of computers which not need to be of high quality but it can be of commodity computers [16], [17]. It has two core components as under [16], [17]:

1) Hadoop distributed file system.
2) Map-Reduce.

1) Hadoop Distributed File System

HDFS is a Hadoop distributed file system and processing is done by map reduce component. HDFS contains interconnected clusters of nodes where files and directories reside. It is a distributed file system and designed for storing very large files which has streaming data access patterns i.e. time taken to read complete data is more important than that to fetch a single record of data [18]. Its clusters are running on commodity hardwares.

Components of HDFS are Name node and Data node. Name node is the master of the system which maintains and manages the blocks which are present in the data nodes. Job Tracker runs on name node and task Tracker runs on data node. Data node can be redundant. It is also called slaves that are deployed on each machine and provide the actual storage and responsible for serving read and write request for client. fig. 1 illustrates HDFS Architecture. In which name Rack is given, that rack is a collection of data node in a cluster. We have a client which interacts with namenode as well as with datanode. The cluster metadata and datanode that stores the data file are managed by Namenode. Contents of file splits into large blocks and independent replication of each block of the file at multiple datanodes are done. Datanode consist of blocks. Replicas of the block are continuously monitored by Namenode. If replica of the block is lost then Namenode creates another replica of that block. Instead of sending direct requests to datanodes, namenodes are sending instructions by replying heartbeats sent by those datanodes. In that instruction, it includes replication of the blocks to other datanodes, remove local block replica, re-register and send an immediate block report or shut down the node [14].

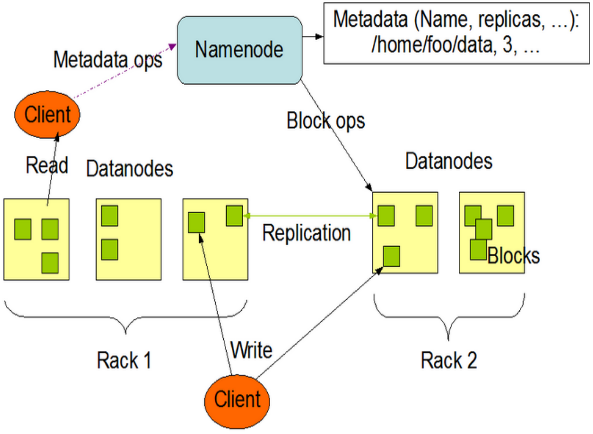

Fig. 1: HDFS Architecture [17]

2) Map Reduce

It is a programming model provided by Hadoop to retrive and analyze data [16]. In traditional system we have to do manualy all the things such as splitting of data into multiple small chunks, aggregate/concate all the context and put on a specific file [16]. Whereas in Hadoop framework with MapReduce it splits the data, ability to run the code on the split block and aggregates the data from multiple places automatically and gives consolidated formats [18].

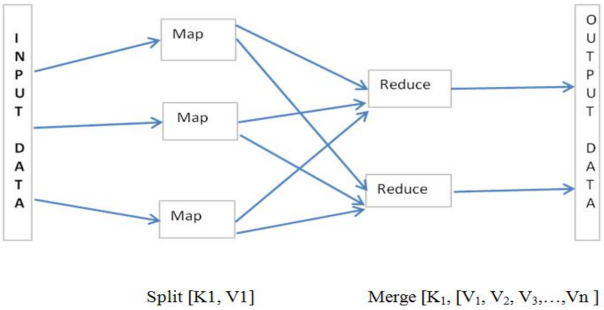

Fig. 2: MapReduce Concept representation [13]

Partition and division of data into multiple small chunks are carried out by Map task. Map( ) procedure performs filtering and sorting. Reduce( ) performs summation operation. Parallel running of various tasks, communication management and data transfers are managed by MapReduce system. By this redundancy, fault tolerance and overall management of the whole process is achieved. Split and merge operations are written as split [K_1, V_1] and merge [K_1, [V_1, V_2, V_3,…,V_n] respectively. In general, data structure can be written as [Key, Value] for both Map and Reduce methods. Applications of MapReduce are weather forecasting, Oil and gas industry and Health care which requires predictive analysis [13], [15].

If failure of one node occurs then Hadoop can easily handle it. So, in [14] they implemented P-PIC algorithm in Hadoop environment. Because of Hadoop framework facility of creating replica there is no loss of data and they got good clusters. Also in [14] they showed that time was decreasing in Parallel Power Iteration Clustering using MapReduce in Hadoop as compared to Parallel Power Iteration Clustering as they were increasing number of processors.

VI. Conclusion And Future Work

Distributed data mining is one of the important aspects of data mining. There are many traditional methods of clustering, among them power iteration method is fast and scalable method. It reduces computational cost but it cannot handle the large datasets so new method is developed called Parallel Power Iteration Clustering. It minimizes communication costs. Due to parallel approach of this method various nodes are created. So, there are chances of failure of such nodes which results in data loss. Therefore to overcome failure of nodes and ultimately data loss, Parallel Power Iteration Clustering with MapReduce in Hadoop framework in distributed environment is invented. In P-PIC with Hadoop, replicas are created.

In future we can use another approach for clustering the data and check performance by using it in Hadoop framework. Problem of fault tolerance overcomes in Hadoop, but if Namenode fails which creates replica then how to take care of data. How fault tolerance can be compared using MapReduce and with other frameworks that can be addressed.

References

[1] Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, University of Illinois at Urbana-Champaign, Elsevier, 2006.

[2] Li Zeng, Ling Li, Lian Duan, Kevin Lu, Zhongzhi Shi, Maoguang Wang, Wenjuan Wu, Ping Luo, ―Distributed data mining: a survey,‖ Springer Science+Business Media, LLC, pp. 304-309, 2012.

[3] Yongjian Fu, ―Distributed Data Mining: An Overview ,‖ Department of Computer Science, University of Missouri-Roll.

[4] K. Kameshwaran, K.Malarvizhi, ―Survey on Clustering Techniques in Data Mining,‖ International Journals of Computer Science and Information Technologies, pp.2272-2276, 2014.

[5] F. Lin, W.W. Cohen, ―Power iteration clustering,‖ 27th International conference on machine learning, pp. 655-662, 2010.

[6] Weizhong Yan, Umang Brahmakshatriya, Ya Xue, Mark Gilder, Bowden Wise ―p-PIC: Parallel power iteration clustering for big data,‖ J. Parallel Distrib.Comput., pp.352–359, 2013.

[7] A. K. jain, M. N. Murty, P. J. flynn, ―Data clustering: a review,‖ ACM computing surveys, pp. 264-323, 1999.

[8] Alex Gittens, Prabhanjan Kambadur, Christos Boutsidis, ―Approximate Spectral Clustering via Randomized Sketching,‖ IBM Research, summer 2012.

[9] W. Kim, ―Parallel clustering algorithms: Survey,‖ CSC 8530 Parallel Algorithms, 2009.

[10] Z. Du, F. Lin, ―A novel parallelization approach for hierarchical clustering,‖ Elsevier, pp. 523-527, 2005.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu**,** ―A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,‖ Proceedings of the Second International Conference on Knowledge Discovery and Data mining, KDD-96, 1996.

[12] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang, ―Parallel Spectral Clustering in Distributed Systems,‖ IEEE Transactions on Pattern analysis and machine Intelligence, pp.568-586, 2011.

[13] D. Jayalatchumy, P. Thambidurai, A. Alamelu, Vasumathi, ―Parallel processing of Big Data using Power Iteration Clustering over MapReduce,‖ IEEE World Congress on Computing and Communication Technologies, pp. 176-178, 2014.

[14] Ankit Darji, Dinesh Waghela, ―Parallel Power Iteration Clustering for Big Data using MapReduce in Hadoop,‖ International Journal of Advanced Research in Computer Science and Software Engineering, pp. 1357-363

[15] D. Jayalatchumy, P. Thambidurai, ―Implementation of P-PIC algorithm in Map reduce to handle big data,‖ International Journal of Research in Engineering and Technology, pp. 113-118, 2014.

[16] http://www.edureka.in/big-data-and-hadoop.

[17] http://hortonworks.com/hadoop/hdfs

[18] “An introduction to Hadoop Distributed File System”, www.ibm.com/developerworks/library/wa-introhdfs, 2014.