Client Based Power Iteration Clustering Algorithm to Reduce Dimensionality in Big Data

Jayalatchumy. D[1], Thambidurai. P [2]
[1, 2] Department of CSE, PKIET, Karaikal, India

Abstract - *Clustering is a group of objects that are similar among themselves but dissimilar to objects in other clusters. Clustering large dataset is a challenging task and the need for increase in scalability and performance formulates it to use parallelism. Though the use of Big Data has become very essential, analyzing it is demanding. This paper presents the (pC-PIC) parallel Client based Power Iteration clustering algorithm based on parallel PIC originated from PIC (Power Iteration Clustering). PIC performs clustering by embedding data points in a low dimensional data derived from the similarity matrix. In this paper we have proposed a client based algorithm pC-PIC that out performs the job done by the server and reduces its execution time. The experimental results show that pC-PIC can perform well for big data. It's fast and scalable. The result also shows that the accuracy in producing the clusters is almost similar to the original algorithm. Hence the results produced by pC-PIC are fast, scalable and accurate.*

Keywords: PIC, p-PIC, pC-PIC, Big Data, Clustering.

1 Introduction

Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. Survey papers, e.g., [1][2] provide a good reference on clustering methods. Sequential clustering algorithms work well for the data size that is less than thousands of data sets. However, the data size has been growing up very fast in the past decade due to the rapid improvement of the information observation technology.

The characteristics volume, velocity and variety are referred to as big data by IBM. Big data is used to solve the challenge that doesn't fit into conventional relational database for handling them. The techniques to efficiently process and analyze became a major research issue in recent years.

One common strategy to handle the problem is to parallelize the algorithms and to execute them along with the input data on high-performance computers. Compared to many other clustering approaches, the major advantage of the graph-based approach is that the users do not need to know the number of clusters in advance. It does not require labeling data or assuming number of data clusters in advance. The major problem of the graph-based approach is that it requires large memory space and computational time while computing the graph structure [5]. The limitation comes from computing the similarity values of all pairs of the data nodes

Moreover, all the pairs must be sorted or partially sorted since the construction of the graph structure must retrieve the most similar pair of the data nodes. This step is logically sequential and thus hard to be parallelized. Unfortunately, this step is necessary and it takes most of the computational time and memory space while performing clustering. Therefore, to parallelize the graph-based approach is very challenging. One popular modern clustering algorithm is spectral clustering. Spectral clustering is a family of methods based on Eigen decompositions of affinity, dissimilarity or kernel matrices [5][7].

PIC replaces the Eigen decomposition needed by spectral clustering with matrix vector multiplications, which can reduce computational complexity. By performing clustering on several datasets it has been proved that PIC [5] is not only accurate but also fast. The PIC algorithm can handle large data's but fitting the similarity matrix into the computer's memory is

not feasible. For these reasons we move on to parallelism across different machines. Due to its efficiency and performance for data communications in distributed cluster environments, the work was done on MPI as the programming model for implementing the parallel PIC algorithm

2 Power Iteration Clustering

Spectral clustering has its own advantages over other conventional algorithms like K-means and hierarchical clustering. The use computing eigenvector is time consuming [5]. Hence PIC is designed to find pseudo-eigenvector thus it can overcome the limitation.

The effort required to compute the eigenvectors is relatively high, O(n3), where n is the number of data points. PIC [9] is not only simple but is also scalable in terms of time complexity O(n) [5]. A pseudo-eigenvector is not a member of the eigenvectors but is created linearly from them. Therefore, in the pseudo Eigen vector, if two data points lie in different clusters, their values can still be separated.

Given a dataset X = (x_1; x_2…..x_n), a similarity function s(x_i; x_j) is a function where s(x_i; x_j) = s(x_j ; x_i) and s >= 0 if i ≠ j, and s = 0 if i = j. An affinity matrix A€ R^{nxn} is defined by A_{ij} = s(x_i; x_j). The degree matrix D associated with A is a diagonal matrix with d_{ii} = $\sum_{ij}$ Aij: A normalized affinity matrix W is defined as $D^{-1}A$. Thus the second-smallest, third-smallest,. . . , k^{th} smallest eigenvectors of L are often well-suited for clustering the graph W into k components[10].

The main steps of Power iteration clustering algorithm are described as follows [9] [5]:

1) Calculate the similarity matrix of the given graph.
2) Normalize the calculated similarity matrix of the graph, $W=D^{-1} A$.
3) Create the affinity matrix A € R^{n*n} W from the normalized matrix, obtained by calculating the similarity matrix.
4) Perform iterative matrix vector multiplication is done V^{t+1}
5) Cluster on the final vectors obtained.
6) Output the clustered vectors.

Input: A data set x={x_1, x_2,….x_n} and normalized affinity matrix W

//Affinity matrix calculations and normalization

1. Construct the affinity matrix from the given graph, A € R^{n*n}
2. Normalize the affinity matrix by dividing each element by its row sum, $W=D^{-1} A$.

//Steps for iterative matrix- vector multiplication

repeat

$V^{t+1} = (WV^t)/||WV^t||_1$
$\sigma^{t+1} = |V^{t+1} - V^t|$
Acceleration=$||\sigma^{t+1} - \sigma^t||$
Increase t
Until stopping criteria is met.

//Steps to generate initial vector
Generate initial vector, $V_o = R/||R||_1$ where R
is the row sum of W.
Steps for clustering
Cluster on the final vector V^t.
Output the clusters.

Fig 1: Pseudo code for PIC

The affinity matrix is,

$$\begin{bmatrix} 4 & 2 & 2 \\ 2 & 5 & 3 \\ 2 & 3 & 5 \end{bmatrix}$$

The normalized row sum for the first row is 4.9

$$W = \begin{bmatrix} 0.8 & 0.4 & 0.4 \\ 0.3 & 0.8 & 0.4 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$$

The row sum R is calculated as,

$$\begin{bmatrix} 1.5 \\ 1.6 \\ 1.0 \end{bmatrix}$$

Hence the normalized matrix is ||R||= 2.41. The value of V_o is calculated as 0.414.When t=0, V^1 is obtained from the following $V^1=WV^0/||WV^0||$ Hence,

$$V^1 = \begin{bmatrix} 0.65 \\ 0.61 \\ 0.40 \end{bmatrix}$$

After all the necessary calculations, V^1 and V^0 after substitution produce zero, hence we conclude that the number of clusters produced is two. The experimental result of implementation of PIC algorithm for various input and various clusters generated for the given inputs are shown as graphs in the fig 2 given below.

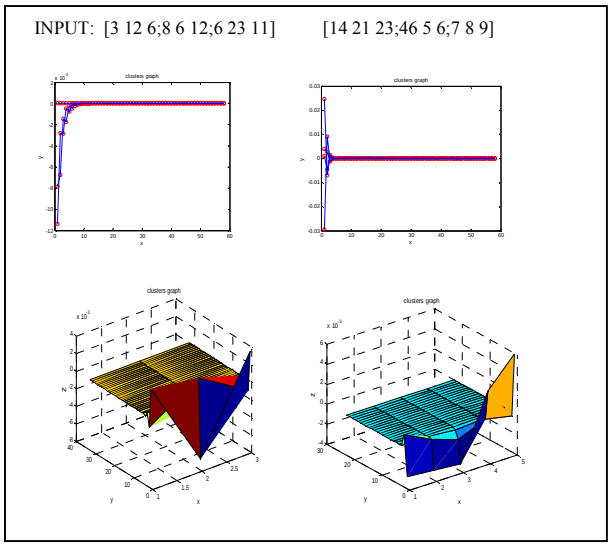

Fig 2.The graph for various inputs

3 Parallel PIC (p-PIC)

Parallelization is a method to improve performance and achieve scalability. Many techniques have been used to distribute the load over various processors. There are several different parallel programming frameworks available [12]. The message passing interface (MPI) is a message passing library interface for performing communications in parallel programming environments [12]. Because of the efficiency and performance on a distributed environment, work has been done on MPI [5] as the programming model and implemented the parallel PIC algorithm. The algorithm for parallel PIC is as follows [5] and the flowchart is shown in fig 3.

Step 1: Get the starting and end indices of cases from master processor.
Step 2: Read in the chunk of case data and also get a case broadcasted from master.
Step 3: Calculate similarity sub-matrix, A_i ,a n/p by n matrix.
Step 4: Calculate the row sum, R_i , of the sub-matrix and send it to master.
Step 5: Normalize sub-matrix by the row sum, $W_i = D_i^{-1} A_i$.
Step 6: Receive the initial vector from the master, v^{t-1}.
Step 7: Obtain sub-vector by performing matrix-vector multiplication, $v_i^t = \square W_i v^{t-1}$.
Step 8: Send the sub-vector, v_i^t , to master.

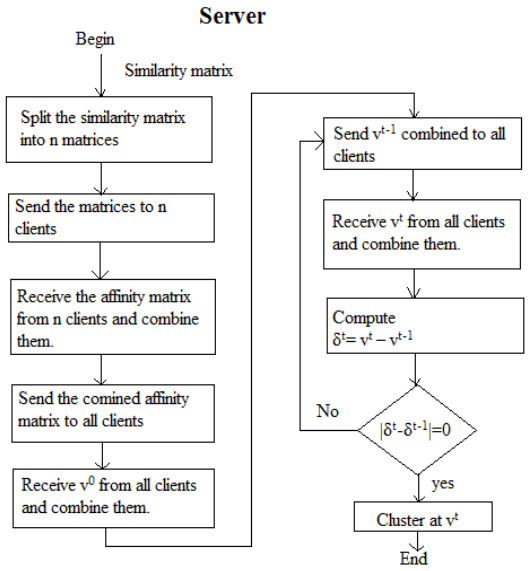

Fig 3: Flowchart for p-PIC using MPI

4 Client Based p-PIC

The time taken for transferring the data from the server to client takes much of the time for execution. Since initial vectors has to be calculated and sent to the master each time to find the vectors, it consume more time. To reduce this process time the algorithm is designed in such a way that the client takes the responsibility of handling much work reducing work of server. The algorithm of client based power iteration clustering is as follows. The master receives the data from the dataset. The data are spilt based on the number of slaves (n). On receiving the data from the master, each slave starts its work of computation. Each slave receives the data file from the master and finds the row sum .The row sum is sent back to the master. Now the master finds the initial vector which is sent to the slave. The calculation of initial vector and number of clusters is calculated and the process ends when the stopping criteria is met. The architecture for the parallel client based PIC flowchart is given below in fig 4.

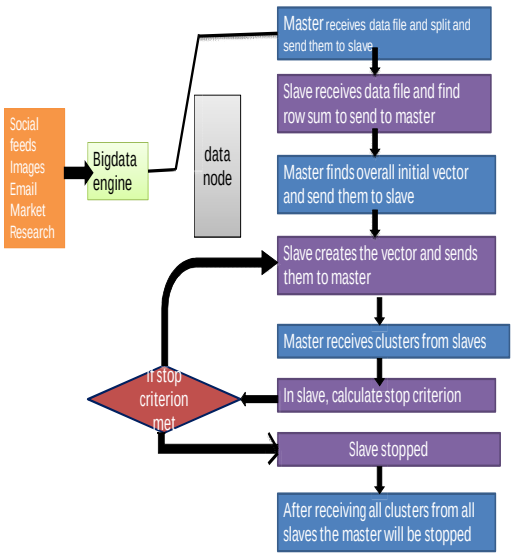

Fig 4: Working of pC-PIC algorithm

5 Experimental Results

The effectiveness of the original PIC for clustering has been discussed by Lin and Cohen [9]. The scalability of p-PIC have been shown by Weizhong Yana [5]. In this paper will focus on scalability in parallel implementation of the pC-PIC algorithm. We implemented our algorithm over a number of synthetic dataset of many records. We also created a data generator to produce a dataset used in our experiment. The size (n) of the dataset varies from 10000 to 100000 numbers of rows. We performed the experiment on local cluster. Our local cluster is HP Intel based and the number of nodes is 6. The stopping criteria for the number of clusters created are approximately equal 0. In this paper we used speed up (execution time) as the performance measure for implementing the pC-PIC.

6 Comparisons of PIC, P-PIC and Pc-PIC

We present performance results of pC-PIC in terms of speedups on different no of processors and the scalability of algorithm with different database size are found .We compare p-PIC with pC-PIC and have shown that the performance have been increased along with the scalability. Fig 5and 6 shows the time executed for various size of datasets. The data sizes are measured in MB and the time taken is calculated in milliseconds. The graph gives a comparison of PIC, p-PIC and p-PIC in MapReduce framework. Fig 7 gives the comparison of various datasets and its corresponding execution time.

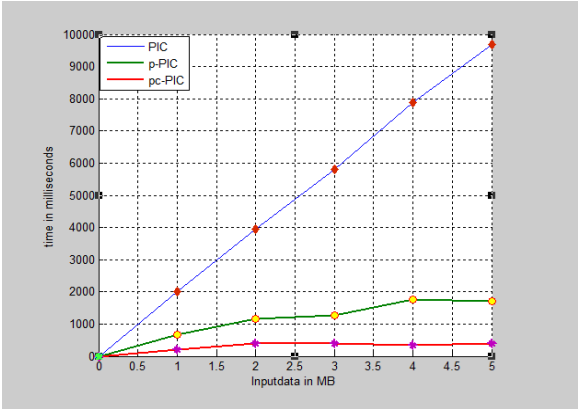

Fig 5: Comparison of Speedup for PIC, p-PIC and p-PIC in MapReduce

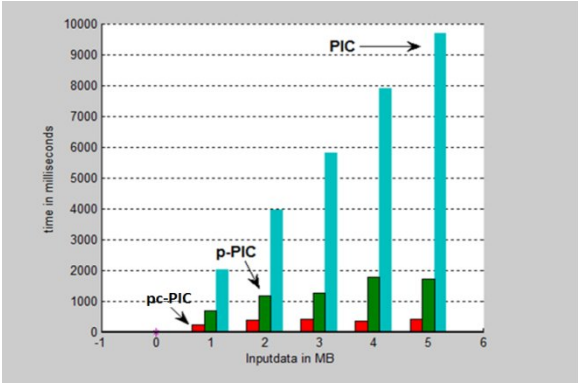

Fig 6: Data (MB) vs Execution time (ms)

Dataset Size (KB)	PIC (ms)	p-PIC (ms)	Pc-PIC (ms)
1000	2012	672	218
2000	3947	1189	385
3000	5838	1259	399
4000	7879	1760	406

Fig 7: Comparison of PIC, p-PIC and pC-PIC

7 Conclusion

In this paper we have designed a new client based algorithm for PIC namely the pC-PIC and have generated cluster for dataset of various size. The results have been compared with sequential PIC and p-PIC using MPI. The results show that the clusters formed using pC-PIC is almost same as that of the other algorithms. The performance has been increased to a greater extend by reducing the execution time. It has also been observed the performance increases along with the increase in the data size. Hence it is more efficient for high dimensional dataset.

8 Future Work

Detecting the failure node that crashes the entire system is necessary. The aim of fault tolerance system is to remove such nodes which cause failures in the system [8]. Using Hadoop the problem of fault tolerance can be avoided. As a future work we can address how node failures can be avoided using Mapreduce and can be compared with other frameworks. Hadoop is fault tolerant and it also provides a mechanism to overcome it.

9 References

[1] Jain, M. Murty, and P. Flynn, "Data clustering: A review", ACM Computing Surveys 31 (3) (1999) 264–323.

[2] Xu, R. and Wunsch, D. "Survey of clustering algorithms", IEEE Transactions on Neural Networks 16 (3) (2005).

[3] Kyuseok Shim, "MapReduce Algorithms for Big Data Analysis", Proceedings of the VLDB Endowment, Vol. 5, No. 12,Copyright 2012 VLDB Endowment 21508097/12/08.

[4] Chen,W.Y , Song, Y, Bai,H. and Lin. C, "Parallel spectral clustering in distributed systems", IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (3) (2011) 568–586

[5] Weizhong Yana et.al , " P-PIC: Parallel power iteration clustering for big data", Models and Algorithms for High-Performance Distributed Data Mining. Volume 73, Issue 3, March 2013

[6] W. Zhao, H. Ma, Q. He, "Parallel K-means clustering based on MapReduce", Journal of Cloud Computing 5931 (2009) 674–679.

[7] H. Gao, J. Jiang, L. She, Y. Fu, "A new agglomerative hierarchical clustering algorithm implementation based on the map reduce framework", International Journal of Digital Content Technology and its Applications 4 (3) (2010) 95–100.

[8] Kyuseok Shim, "MapReduce Algorithms for Big Data Analysis", Proceedings of the VLDB Endowment, Vol. 5, No. 12,Copyright 2012 VLDB Endowment 21508097/12/08.

[9] Frank Lin frank, William W.," Power Iteration Clustering",International Conference on Machine Learning, Haifa, Israel, 2010.

[10] F. Lin, W.W. Cohen, Power iteration clustering, in: Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 2010.

[11] Sakai, T. and Imiya, A. "Fast spectral clustering with random projection and sampling", Lecture Notes in Computer Science 5632 (2009) 372–384.

[12] Quinn, M.J. "Parallel Programming in C with MPI and OpenMP", McGraw-Hill,Boston, Mass, UA, 2008