MR-IDBSCAN: Efficient Parallel Incremental DBSCAN Algorithm using MapReduce

Maitry Noticewala
CSE department
Parul Institute of Technology
29, Gopaleshvar Soc. Tadwadi
Rander Road, Surat-395009

Dinesh Vaghela
CSE department
Parul Institute of Technology
Limda, Vadodara, India

ABSTRACT

Incremental DBSCAN is a one of the density based algorithm to find clusters of arbitrary shapes. This algorithm is one the method of the DBSCAN algorithm. DBSCAN stands for the Density Based Spatial clustering of Application with Noise. This Algorithm find clusters in arbitrary shapes, size, and as well as filter out noise. Various algorithms are invented to improve DBSCAN algorithm in many different ways like time complexity, efficiency, performance. In this research such algorithm will be develop that can work in the distributed environment using the Apache Hadoop and MapReduce that will reduce time of the existing algorithm and dataset from the different site will work together from the single node and find the appropriate result in the distributed environment.

General Terms

Data Mining, Data Clustering, Hadoop, Map/Reduce

Keywords

DBSCAN, IDBSCAN

1. INTRODUCTION

Data mining is a combination of three main factors: Data, Information and knowledge. Data are the most elementary description of the things, events or the activity and transactions. Information is organized data which have some valuable meaning or some useful data. Knowledge is a concept of understanding information based on the recognized pattern or algorithms that provide the information. Data Mining is a technique of finding valuable knowledge from the large amount of dataset [1][2][3]. Main techniques for data mining are classification and prediction, clustering, outlier analysis, association analysis, evolution analysis.

Clustering is one of the techniques applied on the unsupervised dataset. Different types of clustering methods are hierarchical, partition, Density Based method and Grid based method. DBSCAN is one of the density based method. DBSCAN can find arbitrary shape. This algorithm is also sufficient for the spatial dataset and also for the large dataset. There are many different algorithm invented from the original DBSCAN algorithm [4].

Performing DBSCAN algorithm in the real world application is challenging due to mainly two reasons. First is a increasing large amount of dataset rapidly so single machine user cannot handle it or having trouble to handle using single user. Second is cost of DBSCAN algorithm i.e much higher computation time with respect to other clustering algorithms.

Thus many existing studies try to improve efficiency of the DBSCAN algorithm. For example FDBSCAN [7] can find the arbitrary shape same as DBSCAN algorithm but time complexity of this algorithm is less than the original DBSCAN algorithm. ODBSCAN [8] is also find arbitrary shape but the main different is identical circles are predefine in this method. VDBSCAN [9] is also find arbitrary shape and also can find cluster in varied density. ST-DBSCAN [10] is also find cluster in arbitrary shape and all this methods are more efficient than DBSCAN algorithm.

One other technique of DBSCAN algorithm is Incremental DBSCAN (IDBSCAN) [11] algorithm. Time complexity of this algorithm is less than the original DBSCAN algorithm. In this algorithm initially clusters are formed using DBSCAN algorithm and then incremental DBSCAN algorithm will be apply to the new upcoming data.

In this Research Paper real challenge is to perform IDBSCAN algorithm in distributed environment using Hadoop open source platform and Map/Reduce framework. Here Map/Reduce algorithm use for the optimize load balancing, execution efficiency and fault tolerance capacity.

Remain paper is organized as follow. In Section 2 we discuss several clustering methods and as well as several related to DBSCAN algorithm. In section 3 we introduce proposed algorithm that is MR-IDBSCAN algorithm, Section 4 is related to the experimental setup and result analysis, and conclusion and future work is given in the Section5.

2. BACKGROUND AND RELATED WORK

Density Based algorithm is one of the approach for the clustering algorithm. It is mainly based on the core points, border points, and density reachable points. In this approach data which are cover in the denser region will be grouped and form one cluster. They use some threshold value to determine denser region. DBSCAN is one of the density based clustering algorithm with noise. DBSCAN can find cluster in arbitrary shape and filter noise. DBSCAN is not effective in massive and varied dataset. It has lower time complexity. To achieve the problem of the lower time complexity new method invented that is IDBSCAN [11].

Apache-Hadoop [12] is one of the open-source platforms to perform distributed data mining for the big data. Hadoop software library is a framework that allows for the distributed processing of the large dataset across cluster of computers using single programming models. It is designed as way so that from the single servers, thousands of machine’s data can be mine in the scalable manner. Each local machine offering computation and storage, Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. Hadoop is combination of two main components. First is a HDFS that is Hadoop distributed File System and second is the Map/Reduce.

2.1 IDBSCAN: Incremental DBSCAN algorithm [11].

IDBSCAN algorithm is capable of adding points in the bulk to existing set of clusters. In this algorithm data points are added to the first cluster using DBSCAN algorithm and after that new clusters are merged with existing cluster to come up with the modified set of the clusters. In this algorithm clusters are added incrementally rather than adding points incrementally. R*- tree data structure is use in this algorithm. In this algorithm new data points which intersect with old data points are determine. For each intersection point, new dataset use incremental DBSCAN algorithm to determine new cluster membership. Cluster memberships of the remaining points are then updated.

Here existing clusters are referring as old cluster and cluster points added are referring as new clusters. By adding the new data points following transitions are possible. In this case Eps and Minpts will be same.

It may possible that old Noise points may become border point or core point in the new cluster formation. Border point in the old cluster may become core point in the new cluster and core point of old cluster may become core point of the new cluster.

Case 1: The affected point to see if any point is density reachable from core point of old cluster, cluster membership is changed and it become core point of the new cluster.

Case 2: If the affected points are density reachable from the core point of the old cluster than these two clusters will be merged.

Case 3: If border point of old cluster becomes core point of new cluster than two clusters will merge. If the border point is not becoming a core point, then it retains its cluster membership. In all the above cases, if the new point is not an intersection point, then its cluster membership will not change.

Case 4: If any point becomes a border point of a new cluster, then it will be absorbed in the cluster. If it becomes a core point, then formation of a new cluster or merging of clusters may happen.

2.2 Map/Reduce overview [13].

Hadoop Map Reduce is a software framework for easily writing applications which process vast amounts of data in-parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner.

Map/Reduce mainly combination of two components. Map function and reduce function. They are defined as follow:

Map: (K1, v1) -> (K2, v2) and Reduce: (K2, v2) -> (K2, v3)

Map function is containing of list of key/value pair and output a list of intermediate key/value pair. Reduce function take that intermediate key/value pair associate with the same key and produce list of key/value pair. The sorted output is a final output of map/reduce process.

A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The MapReduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

3. PROPOSED ALGORITHM

3.1 Overview of proposed work

In the proposed work I am planning to work with incremental DBSCAN algorithm in the Distributed environment using Hadoop. In the proposed system data are spatial dataset. Incremental DBSCAN algorithm will be apply on that different site individually and create one local cluster call it as LC. This LC will be send to the Master Node of the Hadoop System from the entire site. At the Master Node one global cluster (GC) will be generated having the entire Local clusters (LC). Master Node taking job from the different site also called as client. If any client sends any request to the master node master node send back the data node list to the client and then client will communicate with that particular data node. At the data node that query or request will be executed with the help of the map reduce function and that result will be send back to the client.

Following figure is shows the method for the incremental DBSCAN algorithm applies on the distributed system.

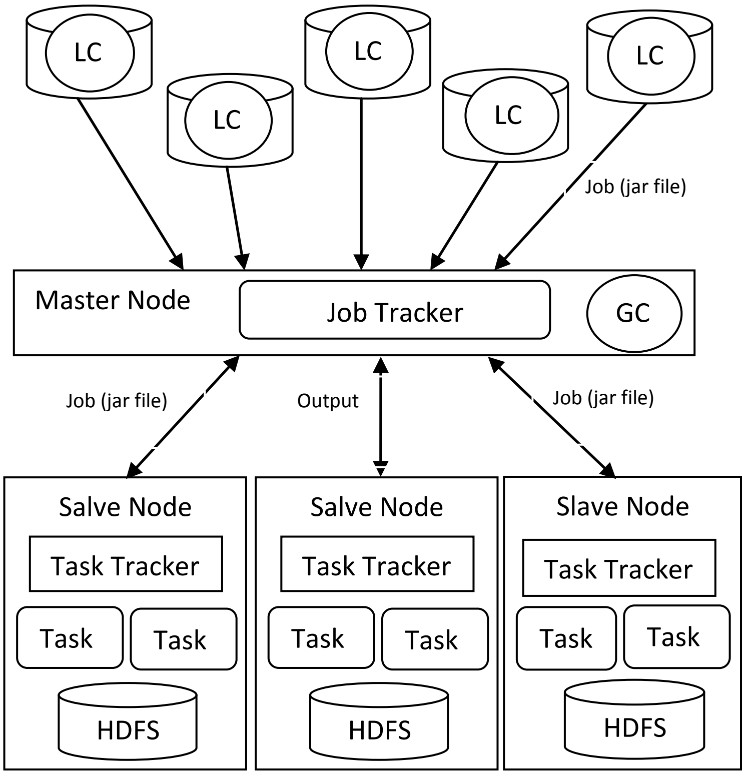


Fig. 1: Incremental DBSCAN algorithm in Distributed Environment

3.2 Algorithm of Proposed Work

Following method will be applied to the System:

(i) In the initial phase all the Distribute site will apply DBSCAN algorithm at the local Database.

(ii) Clusters will be generated at the local sites that clusters are called as LC.

(iii) Above method will be applied to the all different sites.

(iv) This all LC will be send to the Master Node and create one Global cluster called GC and Job Tracker will send that clusters to the data node and store in HDFS file.

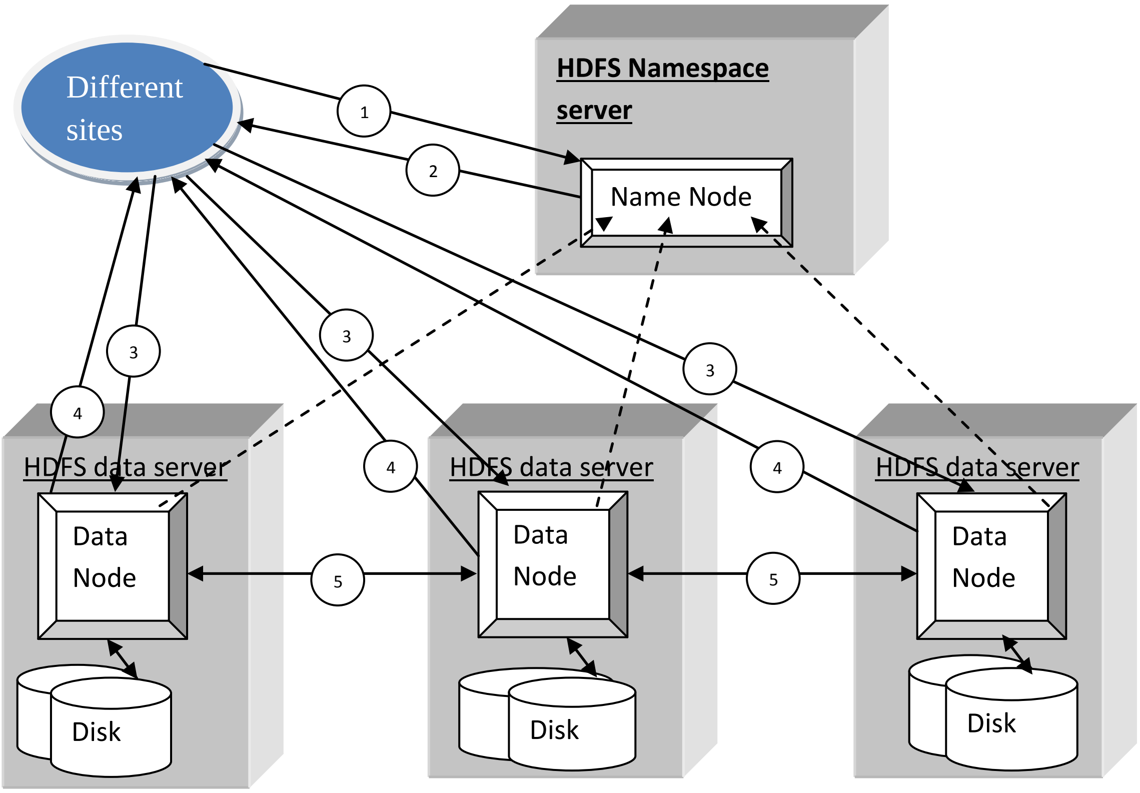
(v) When new data come to the different site client will ask master node for the list of data node.

(vi) Than from the list client will write new data with noise of previous phase to the data node in the HDFS file.

(vii) Data node will process new data with the existing cluster.

(viii)In that process either cluster will be merge with the existing cluster or new clusters will be create.

(ix) When client want to fetch any data than query will be send to any data node and output will be generated at the data node and send back to the client.



1- client application Read , write , locate request to name node
2- Name node reply: list of server and relevant data blocks
3- client application data block manipulation request
4- name node reply: block data if read status
5- data node to data node block replication of data
- - -▶ - Heartbeat and status update from data node to name node

Fig. 2: HDFS structure use by Incremental DBSCAN algorithm

4. EXPERIMENTAL SETUP AND RESULT ANAYSIS

In the proposed system dataset are distributed among the different sites that means data are spatial dataset. In the initial phase clusters will be generated at different site using DBSCAN algorithm. After that all that clusters will be send to the Master Node of the Hadoop system. In the master node all that clusters will be send to the 3 slave node as it is feature of the Hadoop that it can generate 3 replicas of data by default. Noise in the dataset will remove at individual site only in the initial phase and store in .csv file.

When new data will arrive at individual site, it first asks the Name Node to choose Data Nodes to host replicas of the first block of the file. The client organizes a pipeline from node-to-node and sends the new data with previous noise list. When the first block is filled, the client requests new Data Nodes to be chosen to host replicas of the next block. Incremental concept will be apply to the data node only and it will generate new cluster if require or data will be merge with the original clusters.

That final result will be sent back to the client. If any client wants to fetch data from the different data node than it will send query to the master node and that query will be send to data node and process at that data node and it will generate result from all the data node and send result back to the client.

In the Center Based IDBSCAN algorithm main disadvantages is when initially cluster formed using DBSCAN algorithm at that point noise will be remove here in the MR-DBSCAN algorithm noise of initial phase cluster will be added to the next phase process and that noise might be possible that it added to new cluster with new data or merge with existing cluster.

5. PERFORMANCE ANALYSIS

The Basic DBSCAN algorithm, IDBSCAN algorithm in central data mining and proposed MR-IDBSCAN algorithm are implemented in JAVA language using Eclipse IDE. Proposed algorithms tested on Ubuntu 32-bit using Hadoop setup. Here for the algorithm basic input parameter Minpts is taken as 6 and Epsilon is taken as 0.9.

In the below table, results comparison is done between running time, number of clusters generated and loss of objects for n=1000, n=5862, n=7999 and n=13681. Here by the use of different charts it is shown that the time taken by the MR-IDBSCAN is comparatively less than the other two methods and also shown clusters generated by the different method for the different numbers of objects.

Table1. Running Time of different algorithm in Minute

Number of Objects	DBSCAN			IDBSCAN			MR-IDBSCAN		
	Running Time	Number of clusters	Object loss	Running Time	Number of clusters	Object loss	Running Time	Number of cluster	Object loss
1000	8.14	12	223	15.51	23	148	0.12	25	130
5862	5.05	8	52	4.61	15	55	2.3	16	59
7999	13.24	8	38	8.02	12	50	4.33	15	55
13861	65.86	10	37	24.34	11	37	11.67	14	35

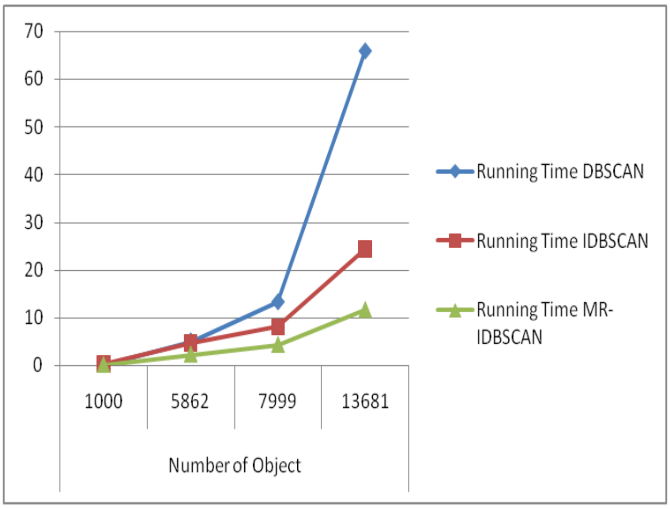


Fig.3: Time Comparison for different objects

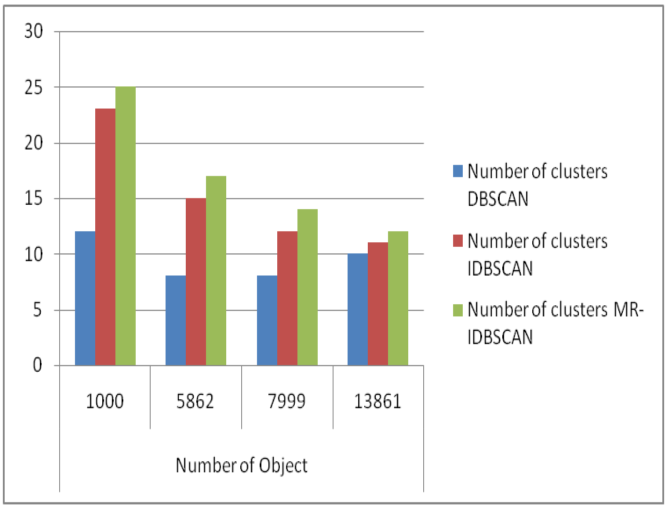


Fig.4: Clusters Comparison of different method

6. CONCLUSION

Compare to central data mining clustering techniques, Distributed data mining is more efficient, scalable and performance is better than the central data mining techniques. Incremental DBSCAN algorithm is better than the other method of the DBSCAN. Incremental DBSCAN can give better performance in distributed environment in terms of run time complexity using Hadoop platform we can reduce performance evolution time and it has also deal with fault tolerance.

In this algorithm it is difficult to delete clusters incrementally from an existing set of clusters. In data warehouses, many times when new data is added during the refresh cycle, old data is purged. So it will be difficult to delete clusters belonging to a particular time period and see its effect on the existing clusters.

7. REFERENCES

[1] Fayyad, Usama; Gregory Piatetsky-Shapiro, and Padhraic Smyth

[2] Han, P.N., Kamber, M.: Data Mining: Concepts and Techniques,2ed(2006).

[3] Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining (2006).

[4] J. Han, M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA, 2001, pp. 335–391.

[5] Khushali Mistry, Swapnil Andhariya, Prof. Sahista Machchhar" NDCMD: A Novel Approach Towards Density Based Clustering UsingMultidimensional Spatial Data". International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 www.ijert.org IJERTIJERT Vol. 2 Issue 6, June - 2013Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[6] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A densitybased algorithm for discovering clusters in large spatial databases," in *Knowledge Discovery and Data Mining*, 1996.

[7] SHOU Shui-geng, ZHOU Ao-ying JIN Wen, FAN Ye andQIAN Wei-ning.(2000) "A Fast DBSCAN Algorithm" Journal of

[8] J. Hencil Peter, A. Antonysamy" An Optimised Density Based Clustering Algorithm" International Journal of Computer Applications (0975 – 8887) Volume 6– No.9, September 2010.

[9] Wei Wang, Shuang Zhou, Bingfei Ren, Suoju He"IMPROVED VDBSCAN WITH GLOBAL OPTIMUM K" ISBN: 978-0-9891305-0-9 ©2013 SDIWC

[10] Derya Birant, and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial-temporal data Data Knowl. Eng. (January 2007)

[11] Navneet Goyal, Poonam Goyal, K Venkatramaiah, Deepak P C, and Sanoop P S" An Efficient Density Based Incremental Clustering Algorithm in Data Warehousing Environment" *2009 International Conference on Computer Engineering and Applications IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore*

[12] http://hadoop.apache.org/

[13] http://www01.ibm.com/software/data/infosphere/hadoop/

[14] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html