iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** XI **Month of publication:** November 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

DATMin-A Novel large Data mining Technique for Concept Evolution and Feature Evolution using Map Reduce Paradigm

Ms. K. Amrita Priya[1], Mrs. D. Priyadarshini[2]
[1]Research Scholar, [2]Assistant Professor, Dept. of Computer Science
Sree Narayana Guru College of Arts and Science

Abstract: ***The Distributed Large Scale data mining using map reduce paradigm is been carried out in large scale using classification and clustering algorithm as it is treated as computational intensive task. The Data Classification and Clustering algorithm using supervised and unsupervised learning models has imposed much impact in map reduce paradigm due to evolution in Concept and feature in the large streaming application. Hence it becomes mandatory to model a system which is capable of handling of the data evolution in the large data. In this Paper, we propose a novel technique named as "DATMin" which develop a learning model in the map reduce paradigm using key value pair processing & Concept based Mining techniques for map and reduce functions. The proposed Technique is capable of handling the diverse computation characteristics such as accuracy and complexity. It will be handled using sophisticated iterative models for feature extraction, classification and feature or concept evolution determination. The Experimental results prove that proposed System outperforms other state of approaches using evaluation metrics such Runtime &Mean Error and reduces I/O overhead to much extent.***
Index Terms – Big Data, Map reduce Paradigm, Key Value Pair, and Incremental Processing

I. INTRODUCTION

The Large data is exploiting day by day which leads to more complexity issues. Big Data is manifested in three different issues. They issues are velocity, Variety and Volume of the data handling and analysing. The data handling and analysis leads different other issues such as integration problem, computation problem, data placement problem, and finally Memory related problem. The Map Reduce paradigm is employed to handle large volume of data with high velocity. The map reduces functions used in production because of its simplicity, generality, and maturity.

Big data is constantly evolving. As new data and updates are being collected, the input data of a big data mining algorithm will gradually change, and the computed results will become more complex and less reliable. The traditional data mining approaches such data classification and clustering which are based on the full batch-mode learning may run short in meeting the demand of analytic efficiency[1]. The Learning model may be supervised or unsupervised. During new data updates arrival to the distributed resource of cluster, data management into the resulting cluster is typical in the data collection process that makes the big data inflate to bigger data, the traditional induction method needs to re-run and the model that was built needs to be built again with the inclusion of new data with evolution in terms of concept and feature. Example of Data Evolving is web pages which grow with new hyperlink about new information about the recent updates. Usually the web page is structured in the graph model to enable efficient searching in the web.

However, the web graph structure is constantly evolving; traditional data mining algorithm will become obsolete. Therefore, it is desirable to propose a new model capable of identifying the data updates with concept evolution and feature evolutions regularly. Hence it becomes mandatory to model a system which is capable of handling of the data evolution in the large data.

Incoop [2] and I2map reduce [3] have used for evolution data processing by extending MapReduce to support incremental processing. However, it has two main limitations. First, Incoop supports only task-level incremental processing. That is, it saves and reuses states at the granularity of individual Map and Reduce tasks. Each task typically processes a large number of key-value pairs (kvpairs). In this Paper, we propose a novel technique named as "DATMin" which develop a learning model in the map reduce paradigm using key value pair processing & Concept based Mining techniques for map and reduce functions. The proposed Technique is capable of handling the diverse computation characteristics such as accuracy and complexity. It will be handled using sophisticated iterative models for feature extraction, classification and feature or concept evolution determination.

©IJRASET: All Rights are Reserved

The rest of paper is organized as follows Section 2 describes the related work, section 3 describes the proposed model in detail, section 4 deals with experimental analysis and finally section 5 is concluded.

II. RELATED WORK

A. *INCOOP- Incremental Processing Approach*

Incoop supports only task-level incremental processing. That is, it saves and reuses states at the granularity of individual Map and Reduce tasks. Each task typically processes a large number of key-value pairs (kvpairs). If Incoop detects any data changes in the input of a task, it will rerun the entire task. While this approach easily leverages existing MapReduce features for state savings, it may incur a large amount of redundant computation if only a small fraction of kv-pairs have changed in a task[4]. Second, Incoop supports only one-step computation. Incoop would treat each iteration as a separate MapReduce job. However, a small number of input data changes may gradually propagate to affect a large portion of intermediate states after a number of iterations, resulting in expensive global re-computation afterwards.

B. *I2MapReduce*

It is an extension to MapReduce that supports fine-grain incremental processing for both one-step and iterative computation. Supports kv-pair level fine-grain incremental processing in order to minimize the amount of re-computation as much as possible. It is modelled as kv-pair level data flow and data dependence in a MapReduce computation as a bipartite graph, called MRBGraph. A MRBG-Store is designed to preserve the fine-grain states in the MRBGraph[5] and support efficient queries to retrieve fine-grain states for incremental processing. It uses reuse the converged state from the previous computation and employ a change propagation control (CPC) mechanism.

C. *Proposed System*

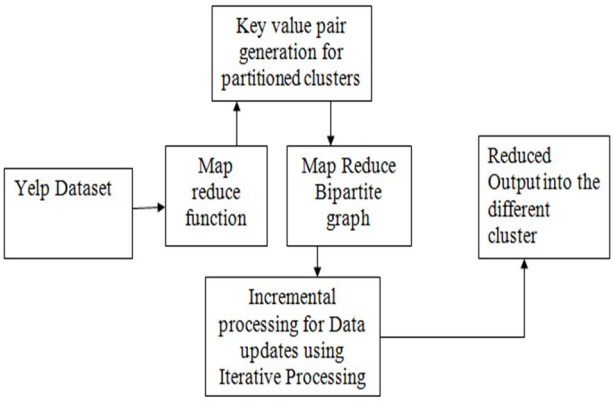
1) *Data Partitioning:* The data is initially partitioned into different clusters and placed in distributed file system using random partitioning algorithm or equal sized blocks. The equal sized partitioning is obtained from the dataset in order to establish the computation using map reduce framework. The Clustered undergoes analysis in the map reduce framework for merging mechanism in the data management.

III. MODELLING A MAP REDUCE ARCHITECTURE

The Map function takes a key value-pair as Input and it computes more Key value pair. The Reduce function is used to produce the final key value pair. The data is placed in cluster in the distributed file system. The Map reduce undergoes following process

A. *Map Function*

For a Map Reduce program, the Map Reduce system runs a Job Tracker process on a master node to monitor the job progress, and a set of Task Tracker processes on worker nodes to perform the actual Map and Reduce tasks. The detailed architecture of the proposed framework is depicted in the figure 1



©IJRASET: All Rights are Reserved

Figure 1 - Architecture Diagram of the proposed System

The Job Tracker starts a Map task per data block, and typically assigns it to the Task Tracker on the machine that holds the corresponding data block in order to minimize communication overhead. Each Map task calls the Map function for every input of each input key value pair, and stores the intermediate key value pair for the redundant and similar contents.

B. Reduce Function

Intermediate results are shuffled to Reduce tasks according to a partition function.Reduce task obtain and merge intermediate results from all Map Tasks.

1) *Data Merging using Map Reduce Bipartite Graph:* Each vertex in the Map task represents an individual Map function call instance on a pair. Each vertex in the Reduce task represents an individual Reduce function call instance on a group. An edge from a Map instance to a Reduce instance means that the Map instance generates key value pair that is shuffled to become part of the input to the Reduce instance. Edges are the fine-grain states M that we would like to preserve for incremental processing[6]. An edge contains three pieces of information: (i) the source Map instance, (ii) the destination Reduce instance, (iii) the edge value. The Map input is the adjacency matrix of the graph. Every record corresponds to a vertex in the graph. For incremental processing, we preserve the fine-grain MRBGraph edge states.
2) *Data Processing for incremental data with Concept Evolution:* It employs the Concept based mining to identify the concept evolution and feature evolution using adjacency matrix. The key value pair is estimated for the merging with already available cluster using map reduce framework. Data indexing [7] is carried out for the key value pair based similarity computation using vertices and edges of the graph. Word similarity is computed using wordnet tool. Indexing is kept for easy retrieval of the data search in the graph structure model[8].

Algorithm – DATmin Incremental processing

Input – Key value pair and Delta input for Increment with Data Evolution
Output – Data merging for Similarity
Process
Map Phase
Select the random point k as Centroid
Map instance of key value for random point
Edge -→weight between the values
Vertices --- values of the each key
Merge ()
For key=1&& key ==2

If (values 1== value 2)
Agg key 1& key 2 as Agg 1
Else
Create a new id agg 2
Reduce Phase
For new evolution
Find (concept similarity)
If (value 1= = value 2)
Merge ( )
Else
Create new data for key

IV. EXPERIMENTAL RESULTS

We implement the prototype named as DATMin in order to support incremental processing. The prototype implements the map reduce and bipartite graph for data merging and data organization into different clusters. Initially one step algorithm is followed in the i2map reduce and DATmin. The size of the dataset is taken in large scale.

Accumulator Reduce optimization in incremental processing is implemented in the reduce Function. For incremental processing,

©IJRASET: All Rights are Reserved

we model a dataset as a delta input from each incremental data.

A. Performance Analysis

DATmin Produces the low processing power with less complexity. The graph structure is constructed for dataset. The i2map reduce and DATmin will return the exact updated result but at the same time prolong the runtime for i2mapreduce. This is because i2MapReduce pays additional cost for accessing and updating the MRBGraph file

Data Structure of the graph often reflects the problem structure and is read -only during computation in the data merging in the map and reduce function. In contrast, state data is the target results of the new data of concept evolution being updated in each iteration by the algorithm. Structure (state) data can be represented by a set of structure (state) kv-pairs.

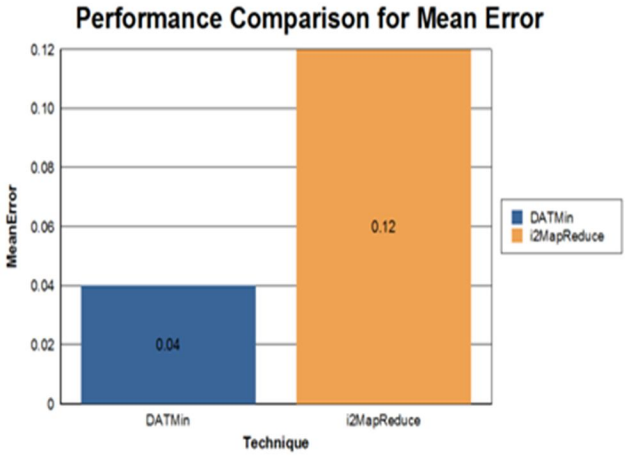


Figure 2: Performance Comparison of Mean error for incremental processing

The DATmin technique speed up convergence, and reduce the time and space overhead for saving states. The Figure 2 describes the Mean error computation of the incremental processing

Further improves the performance with incremental processing of DATmin, reducing the run time as described in the figure 3.

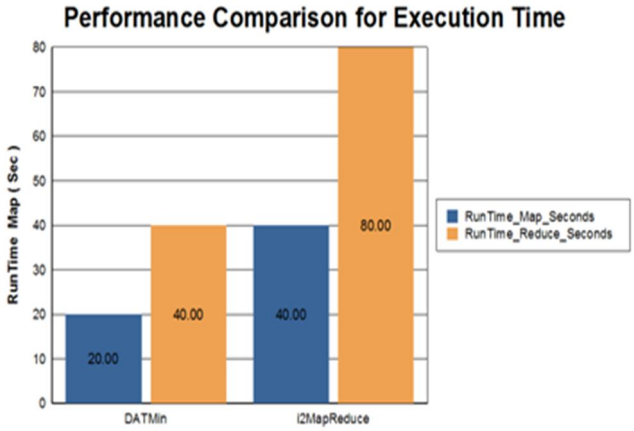


Figure 3- Performance Comparison of the Execution Time against the Incremental processing Techniques

Moreover, we find that the change propagation control mechanism plays a significant role for merging the data. It filters the kv-pairs with tiny changes at the prime. Reduce, greatly decreasing the number of Map instances in the next iteration.

Table 1-Performance Table for the Incremental Processing Techniques

Technique	Run time for Map Function	Run Time for Reduce Function	Mean Error
I2Map Reduce	40s	78s	0.04%
DATmin	20s	43s	0.02%

©IJRASET: All Rights are Reserved

Consequently, the MRBGraph file will consist of multiple batches of sorted chunks, corresponding to a series of iterations. If a chunk exists in multiple batches, a retrieval request returns the latest version of the chunk.The Performance values is depicted in the table 1.

Only the Map and Reduce instances that are affected by the delta input are re-computed. The overhead of the backward transfer can be fully removed if the number of state kv-pairs in the application is greater than or equal to map function. The index is stored in an index file and is preloaded into memory.

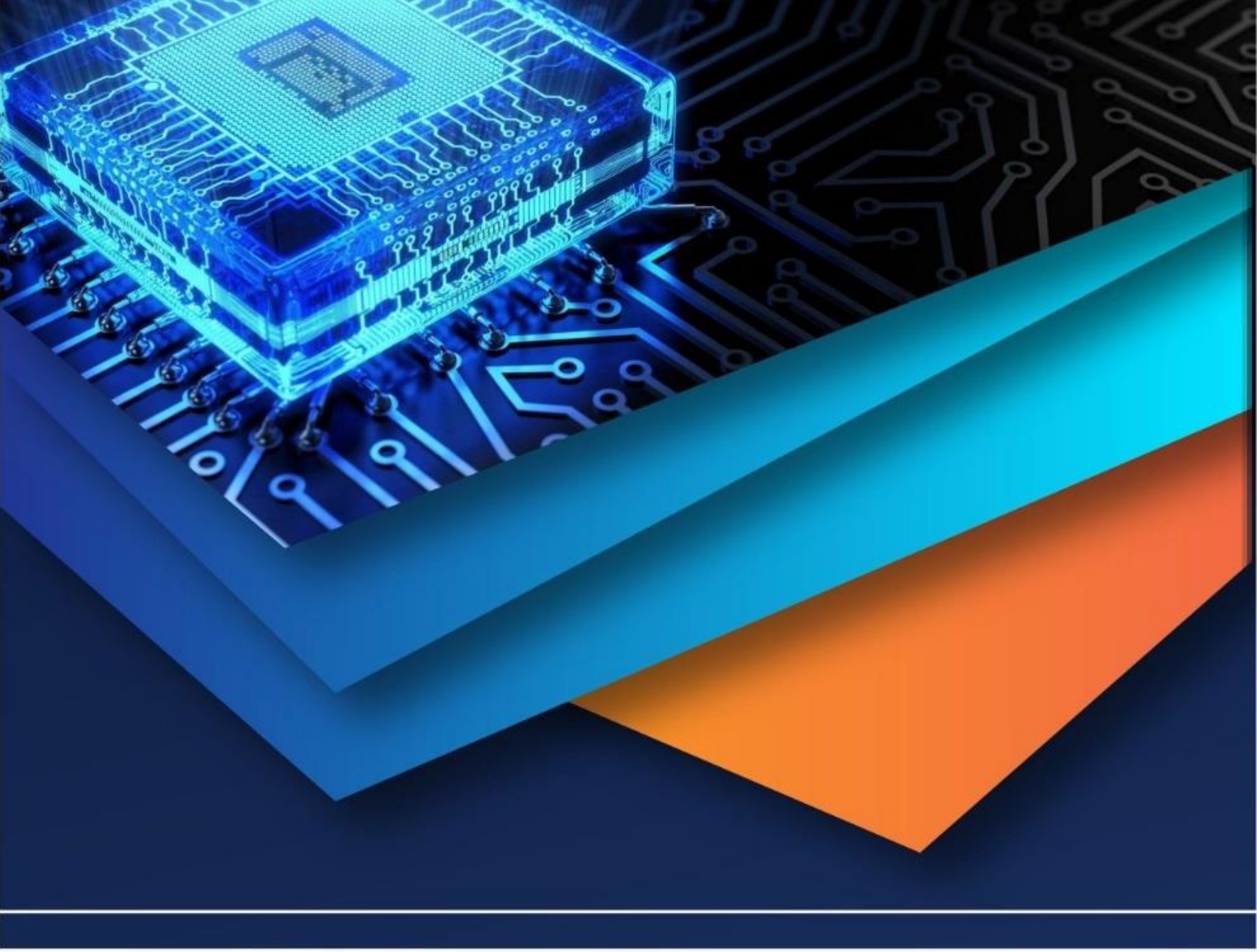
V. CONCLUSION

We designed and implemented the large data processing technique in the map Reduce model for clustering of the data evolutions in the large scale data streaming. Classification model combines unsupervised learning model, concept based mining techniques and Map reduce computation functionalities to generate the clusters of new data updates without re-computation process. Experimental results prove that proposed process significantly reduces the run time and mean error for clustering the data with feature evolution and concept evolutions against re-computation based processes.

REFERENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Opear. Syst. Des. Implementation, 2004, p. 10.
[2] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, "Incoop: Mapreduce for incremental computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 7:1–7:14.
[3] Yanfeng Zhang, Shimin Chen, Qiang Wang, and Ge Yu "i2MapReduce: Incremental MapReduce for Mining Evolving Big Data" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 7, JULY 2015.
[4] T. Y. Wu, J. S. Pan, and C. F. Lin, "Improving accessing efficiency of cloud storage using de-duplication and feedback schemes," IEEE Syst. J., vol. 8, no. 1, pp. 208–218, Mar. 2014, doi:10.1109/JSYST.2013.2256715.
[5] P. Wang, H. Wang, X. Wu, W. Wang, and B. Shi, "A Low- Granularity Classifier for Data Streams with Concept Drifts and Biased Class Distribution," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 9, pp. 1202-1213, Sept. 2007.
[6] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Integrating Novel Class Detection with Classification for Concept- Drifting Data Streams," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 79-94, 2009.
[7] C. Yan, X. Yang, Z. Yu, M. Li, and X. Li, "IncMR: Incremental data processing based on mapreduce," in Proc. IEEE 5th Int. Conf. Cloud Comput., 2012, pp.pp. 534–541.
[8] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "Priter: A distributed framework for prioritized iterative computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 13:1–13:14.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)