International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 7 4880 - 4884

Obtaining Approximation with Data Cube using Map-Reduce

Rohitkumar Kawhale
Computer Engineering Department, G.H.R.C.E.M.
Savitribai Phule Pune University
Pune, India
Rohit.kawhale@gmail.com

Sarita Patil
Computer Engineering Department, G.H.R.C.E.M.
Savitribai Phule Pune University
Pune, India
sarita.patil@raisoni.net

***Abstract*—**Data mining is a field that has an important contribution to data analysis, discovery of new meaningful knowledge, and autonomous decision making. Whereas, the rough set theory offers a viable approach for decision rule extraction from data. With the data cube we tried to put data in multidimensional way and accessed that data via map reduce. The adequate quantity or supply of data, coupled with the need for powerful data analysis tools, i.e. where data is rich but information is in poor situation. The proposed algorithm is been compared with other different rough set approximation approaches. Our algorithm to achieve approximation for decision rules has better performance. This proposed algorithm has been more efficient to obtain approximation.

Keywords- *Data cube; decision table; map; reduce; rough set*

*****

I. INTRODUCTION

Data mining used to extracting knowledge from large amount of data. It discovers interesting knowledge from large amounts of data stored either in data warehouse, database, or other information repositories. The massive data mining is a big challenge. Number of techniques are used to achieve knowledge from raw data. Some of them are fuzzy set, neural network, bays theorem and rough set. If we consider the rough set, the terms lower approximation, upper approximation and the boundary region are very basic and most vital. There are number of ways to calculate these rough approximations. There are number of fields where rough set is used in wide way like medical, engineering, banking, intrusion detection, pattern recognition, quality analysis, artificial intelligence etc. Hadoop is a java framework which is used to store and process large amount of data on commodity hardware. We are able to deal with massive data.

In this paper, we are going to merge two different terms data cube and rough set. By representing given dataset using data cube, we can get rough approximation in easier way. We just need to compare locations of data cube for different decisions. It reduces our task, as number of comparison are not more than domain of decision attribute. Another most important advantage of data cube with rough set is once we initialize data cube to null, and then assign the values. The remaining null value positions gives as possible combinations for the same dataset and we can recommend it to user. Size of data has crossed limits of terabytes, petabytes, Exabyte's also. Hence we are also going to apply our proposed system on map reduce platform. So that we will able to deal with massive data. MAPREDUCE is a parallel distributed programming framework introduced in [14], which can process huge amounts of data in a massively parallel way using simple commodity machines.

II. RELATED WORK

Rough set theory (RST) [15] employed mathematical modeling to deal with class data classification problems, and then turned out to be a very useful tool for decision support systems, especially when hybrid data, vague concepts and uncertain data were involved in the decision process. To use the rough set process, one begins with a relational database, a table of objects with attributes, and attributes values for each object [7]. The goal of structuring decision rules is to enhance the decision-making capability of the knowledge generated with learning algorithms [6] [8]. Most of the traditional algorithms based on rough sets are the serial algorithms and existing rough set tools only run on a single computer to deal with small data sets. It greatly restricts the applications of rough sets. Generally, the computation of approximations is a necessary step in knowledge representation and reduction based on rough sets. To expand the application of rough sets in the field of data mining and deal with huge data sets, one parallel computation of the rough set approximations has computed [2]. It uses four different algorithms to compute equivalence class decision class, Association and indexes. For each algorithms different map reduce were get used. And based on indexes, Rough approximation get calculated. The effective computation of approximation is essential improving the performance of data mining and other related task [9]. MapReduce has been implemented in manage many large-scale computation. The recently introduced MapReduce technique has received much consideration from both scientific community and industry for its applicability in big data analysis [3] [4] [11]. The research works have been carried on performing the cube computation, cube aggregation using the MR framework. Nandi et al. [1] [5] developed a scheme to handle special holistic measures,

 ISSN: 2321-8169
4880 - 4884

III. SYSTEM ARCHITECTURE

The system architecture consist of the sample dataset such that the dataset is been selected then that dataset is been processed from mapper to reducer to create data cube from it. The Null data cube is being created such that it can be used to store object from the data set the figure 1 shows the creation of data cube. After the creation the value according to decision attribute is been stored at data cube. And at the end the reducer will give output as upper approximation, boundary region, lower approximation and NULL. These respective output will be stored in their respective sets.

As the massive data mining is a big challenge. Number of techniques are used to achieve knowledge from raw data. As we are going to use data cube for computing rough approximation, we will deal with decision table.

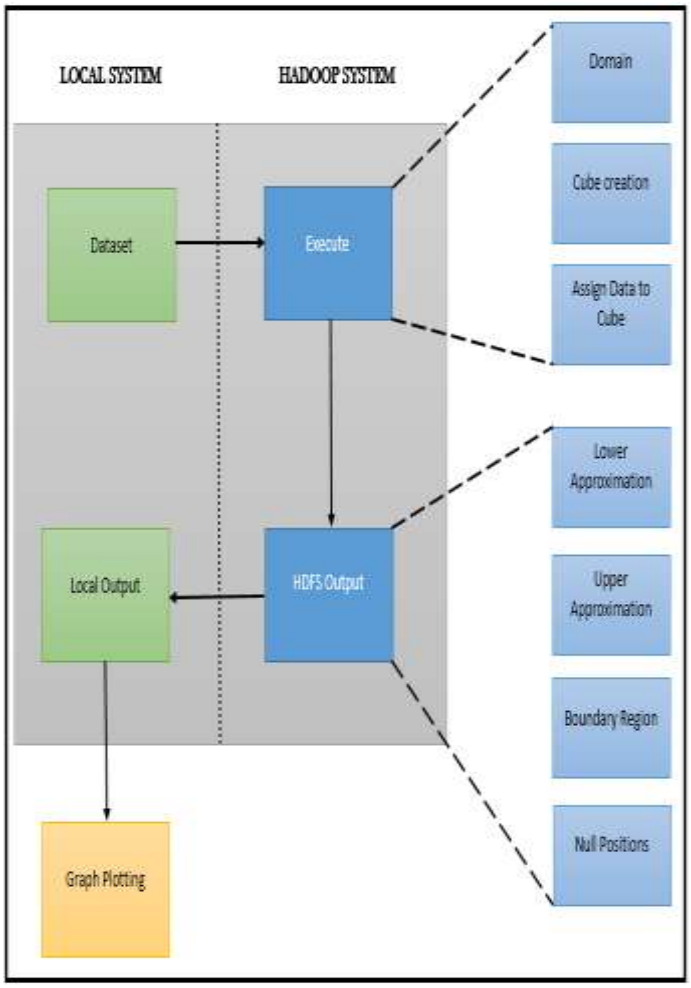

Figure 1: System Architecture

Decision table represented mathematically as,

$$I = \{U, C, D, V\}$$

Where, I is decision table (Information set), U is universe which contains all objects of the decision table. The condition attribute be C and D be the decision attribute of decision table. V contains the values of the information table for every object and its corresponding attribute. It includes domain values of each attribute.

Example: Consider the example as shown in table 1.

TABLE 1

Object	a1	a2	d
x1	0	0	0
x2	0	0	1
x3	1	0	1
x4	1	1	1
x5	0	0	0
x6	1	1	2
x7	0	0	0
x8	1	1	1
x9	1	1	2
x10	0	1	1
x11	1	0	2
x12	1	1	1

We can represent table 1 mathematically as, I={(x1,….x12), (a1,a2), d, ([0,1], [0,1],[0,1,2])}.

A. *Data Cube*

Date cube is a crucial concept and research direction in OLAP (Online Analytical Processing). In former years, the researches on the data cube are focusing mainly on the two aspects: firstly, we can say how to compress the cube and store cube. Becoming to the massive data, storing all the data cubes needs a lot of space and resources. Secondly, how to choose the cube and materialize them. So, in order to help users to get effective data, many learners started paying attention on finding the best method for materializing data cubes. Researchers have suggested a variety of algorithms on Cube compression and Storage, such as quotient cube, star cube, iceberg cube and so on.

A data cube consists of a lattice of cuboids, in that each one corresponding to a different degree of summarization of the given multidimensional data. Even though it is called a 'cube', it can be 2-dimensional, 3-dimensional, or higher-dimensional. Such that, every dimension represents a new attribute in the database and the cells in the cube represent the measure of interest [12].

For any data cube,

No. of attributes (dataset) = no. of dimensions (data cube)

Hence, for table 1, we can generate cube having number of dimensions= 3. i.e. a1, a2, d.

B. *Decision dataset*

Our system mostly work on dataset consisting of decision attribute such that any dataset who has some decision will work fluently with our system. For example we have primarily

 ISSN: 2321-8169
4880 - 4884

consider dataset in table 1 with 12 records only to understand our system easily. The size and record won't matter with our system it will easily provide user the lower and upper approximation.

So from table 1 we can notice that there is conditional attribute with last column as decision attribute. The example dataset consist of three decision as 0, 1, 2 we can select other attribute column as a decision set. Considering last column as a decision attribute and other than that as conditional attribute.

C. Rough Set

Basically rough set is depend on approximation i.e. upper approximation and lower approximation and boundary region as mentioned below which is calculated later in this paper. Approximations are fundamental concepts of rough set theory. Rough set theory is based on three important terms:

- Lower approximation
- Boundary Region
- Upper Approximation

Lower approximation contains the objects which has unique output for the same condition attribute. The objects of the lower approximation generates the rules without any ambiguity.

Unlike the lower approximation boundary region contains the objects which has more than one output for the same condition attribute. The objects of the boundary region generates the rule with ambiguity. Upper approximation is the union of lower approximation and boundary region.

Let T be the decision table and

T = {U, A(C&D), V, f}

where U is universal set and A be the set of attributes consists of condition attribute C and decision attribute D. If B $\subseteq$ A And X $\subseteq$ U, We can approximate X using only the information contained in B by constructing the B-lower and B-upper approximations of X, denoted by $\overline{B}$ and B respectively, where

$$B = \{x|\ [x]_B \subseteq X\}$$
$$\overline{B} = \{x|\ [x]_B \cap X \neq \emptyset\}$$
$$\textit{Boundary Region} = \overline{B} - B$$

IV. Map Reduce

Map-Reduce [8] allows for distributed processing of the Map and Reduce functions [13]. The Map-Reduce divides the input file into no of blocks by "input split" method. Map reduce is used for processing data on commodity hardware.

Figure 2 shows the basic terminology of map reduce initially the given dataset gets split into chunks then its passed to mapper then to reducer. Our system follows this steps of HDFS to achieve the respective output. Initially input has been as text file on which operation has been performed to achieve the results.

The rough set approximations obtained by the parallel method are the same as those obtained by the serial method. But using map reduce we can run independent phases in parallel based on map-reduce. Therefore time required is very less as compared to traditional method of rough set calculation. In addition to that we can also generate the rules for massive data and able to abstract attributes in more efficient way using map-reduce with rough set.

Data partitioning, fault tolerance, execution scheduling are provided by MapReduce framework itself. MapReduce was designed to handle massive data volumes and huge clusters. MapReduce is a java programming framework that allows to execute user code in large cluster. All the user has to write two functions: Map and Reduce. During the Map phase, the input data are distributed across the mapper, where each machine then processes a subset of the data in parallel and produces one or more <key; value> pairs for each data record. Next, during the Shuffle phase, those <key, value> pairs are repartitioned (and sorted within each partition) so that values corresponding to the same key are grouped together into values {v1; v2; ……}. Finally, during the Reduce phase, each reducer machine processes a subset of the <key, {v1; v2; ….}> pairs in parallel and writes the final results to the distributed file system. The map and reduce tasks are defined by the user while the shuffle is accomplished by the system. the map and reduce functions supplied by the user have associated types.

Map (k1, v1) → list (k2, v2)
Reduce (k2, list (v2)) → list (v2)

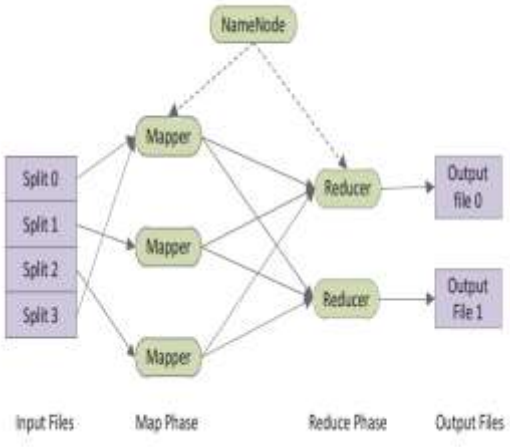

Figure 2: Map Reduce

Two programmer specified functions:

- Map
  Input: key/value pairs i.e. (k1, v1)
  Output: intermediate key/value pairs i.e. list (k2, v2)
- Reduce
  Input: intermediate key/value pairs i.e. (k2, list (v2))

Output: List of values list (v2)

That is, the input keys and values are drawn from a different domain than the output keys and values. The k1, k2 are the two different keys used in MapReduce phase and same as v1, v2 are the different values. The intermediate keys and values are from the same domain as the output keys and values. These keys and values are obtain from the dataset which we browse to process our algorithm and get respective output.

Map-Reduce framework offers clean abstraction between data analysis task and the underlying systems challenges involved in ensuring reliable large-scale computation [10]. Map-Reduce runtime system can be transparently explore the parallelism and schedule components to distribute resource for execution.

V. ALGORITHM

In our proposed system we are going to calculate the rough approximation using data cube.

Basically data cube will be used to represent the dataset. Data cube are an easy way to look at the data. It is used to represent data along some measure of interest. The existing system uses different algorithms to generate equivalent class, decision class, association and indexes to compute the rough approximation. Existing method requires too much computation. And for every algorithms needs map reduce. After studying rough set and data cube, we are going to attempt merge these to concepts to get better performance.

Hence algorithm of our proposed system becomes

Input:

Decision table (data set)

Output:

Upper approximation, Lower Approximation, Boundary region

Method:

Step 1:

Choose the data set

Step 2:

Compute the dimensions of dataset.

No of dimension of data cube = no of attribute (conditions + decision) of decision table

Where,

No. of conditions attributes > 0;

No. of decision attributes = 1;

Step 3:

Initialize the data cube and assign it to NULL

i.e. Values of all positions (indexes) of data cube are initialize to NULL

i) Number of positions of data cube= multiplication of value of each dimension of data cube

Or

ii) No. of positions of data cube = multiplication of domain of condition and decision attribute.

Step 4:

Store the objects of data set to null data cube at corresponding position

i) If more than one objects having same condition attribute value, such objects stored at same position. And treated equally.

ii) Some positions may be remain null for not having value for corresponding position in input dataset.

Step 5:

Compute approximation

i) Compare the objects having same condition but different decision attribute value.

ii) Every comparison, compares objects equal to decision domain.

iii) In comparison if we found for more than one position contains certain value/s, add that objects into boundary region of corresponding decision.

iv) If we found only one value and other positions are null, then add that object into lower approximation of corresponding decision.

v) Upper approximation is union of lower approximation and boundary region.

Step 6:

Add lower approximation, boundary region, upper approximation of all decisions to get lower approximation, boundary region, and upper approximation respectively of input data set

VI. SYSTEM RESULTS

Our system includes different phases initially the cube is been generated from the given dataset. As we have taken preliminary dataset of some record consisting some entries with decision column with referring table 1 so that such data can be used to check the efficiency of our system.

TABLE 2
RESULT

Approximation	Complete Dataset
Lower Approximation	{x10}
Boundary Region	{x1,x2,x3,x4,x5x6,x7,x8,x9,x11,x12}
Upper Approximation	{x1,x2,x3,x4,x5x6,x7,x8,x9,x10,x11,x12}

TABLE 3
RECOMMENDATION NULL POSITIONS

Recommendations	
002	NULL
010	NULL
012	NULL
100	NULL
110	NULL

Similarly preliminary results of our system has been generated shown in table 2. The datasets will be changed for the further work so that more efficient result can be obtained. With the completion of our work the output for the following system is as shown in table 2. Such that table 2 shows over all outcome from the reducer of MapReduce will be generated as for the upper approximation, boundary region, lower approximation and the null positions. Along that table 3 shows all the Null positions as recommendations for the rule generation for that sample dataset.

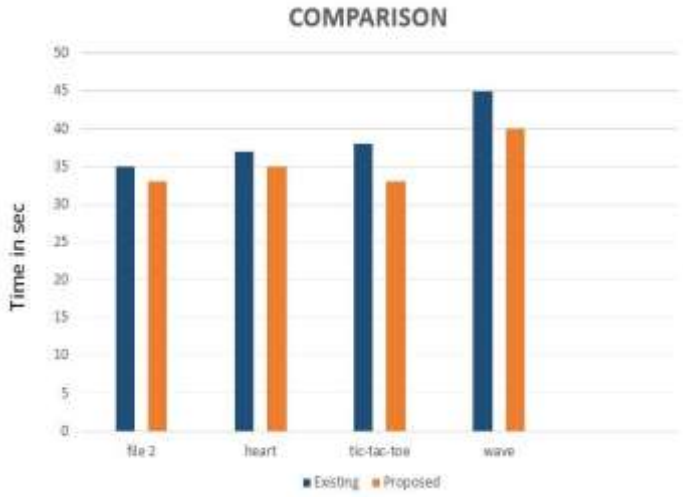

Figure 3 graph of comparison

The figure 3 shows the graph of existing system and preliminary obtain while executing the map reduce on different dataset and time is been calculated and with that the comparison is done between both system. With that we can prove that our proposed system is efficient than the existing system.

VII. CONCLUSION

In this paper the basic concept of roughest and data mining is been discussed. With the previous system there are some of approaches for roughest approximation. With the proposed system we have focus on the data cube roughest with MapReduce. From the data cube the information can be retrieved very easily. So with the proposed system it will be easier to obtain the lower and the upper approximation. Such that, the new algorithm is been designed to work with approximation in one of different way. This algorithm enhances the knowledge from the decision table to the data cube for finding the approximation for the roughest.

REFERENCES

[1] Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan," Data Cube Materialization and Mining over MapReduce", IEEE transactions on knowledge and data engineering, vol. 24, no. 10, October 2012.

[2] Junbo Zhang, Tianrui Li, Da Ruan, Zizhe Gao, Chengbing Zhao "A parallel method for computing rough set approximations" Information Sciences 194 (2012) 209–223.

[3] A. Abouzeid et al., "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads," Proc. VLDB Endowment, vol. 2, pp. 922-933, 2009.

[4] Thomas Jörg, Roya Parvizi, Hu Yong, Stefan Dessloch "Incremental Recomputations in MapReduce" ACM 978-1-4503-0956-1/11/10 October 28, 2011.

[5] A.Nandi, C. Yu, P. Bohannon, and R. Ramakrishnan, "Distributed Cube Materialization on Holistic Measures," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.

[6] Riadh Ben Messaoud, Sabine Loudcher Rabas´eda, Omar Boussaid,Rokia Missaoui "Enhanced Mining of Association Rules from Data Cubes"ACM 1595935304/06/0011 November 10, 2006

[7] Mehran Riki , Hassan Rezaei "Introduction of Rough Sets Theory and Application in Data Analysis" Journal of Mathematics and Computer Science 9 (2014), 25-32.

[8] Andrew Kusiak "Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing" IEEE TRANSACTIONS ON ELECTRONICS PACKAGING MANUFACTURING, VOL. 24, NO. 1, JANUARY 2001.

[9] A.Pradeepa1, Dr. Antony Selvadoss Thanamani "Hadoop File System And Fundamental Concept Of Mapreduce Interior And Closure Rough Set Approximations" International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2013.

[10] Chun Kit Chui, Ben Kao, Eric Lo, Reynold Cheng "I/O-Efficient Algorithms for Answering Pattern-Based Aggregate Queries in a Sequence OLAP System" ACM 978-1-4503-0717-8/11/10 October 24–28, 2011.

[11] Xiaolei Li, Jiawei Han, Zhijun Yin, Jae-Gil Lee, Yizhou Sun "Sampling Cube: A Framework for Statistical OLAP Over Sampling Data" ACM 978-1-60558-102-6/08/06 June 9–12, 2008.
http://en.wikipedia.org/wiki/Hypercube.

[12] Data mining concepts and techniques, Jawai Han, Michelline Kamber, Jiran Pie, MorganKaufmann Publishers, 2nd Edition.

[13] Alberto Abelló Jaume Ferrarons , Oscar Romero "Building Cubes with MapReduce" ACM 978-1-4503-0963-9/11/10 October 28, 2011.

[14] Prachi Patil, "Data Mining with Rough Set Using MapReduce" International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) ISSN(Online) : 2320-9801 Vol. 2, Issue 11, November 2014.