Classification by Rules Mining Model with Map- Reduce framework in Cloud.

SUJAY BASU[#1], Dr. KUMARAVEL[#2]

[#1]*PG Scholar, Bharath University (computer department), India*
A-2/101, kalyani.nadia, westbengal, India, pin-741235
[#2]*Dean, Bharath University (computer science), India*

[1]sujay_basu2003@yahoo.com
[2]drkumaravel@gmail.com

***Abstract*-- Map reduce technique is a good framework for processing and generating a large amount of data. So it can be used for research and application of data mining, for the reason that it provides powerful capacities of storage and computing, excellent resource management based on virtualization and resource sharing model, and comprehensive service system. In this paper, solvent of classification rules mining with resources in cloud is developed, and an innovative classification rules mining model with jrip algorithm with Map reduce technique is proposed. An illustrative example is analyzed to show feasibility and effectiveness of the suggested model.**

***Keywords*—MapReduce,classification,cloudcomputing, mining.**

I. Introduction

MapReduce is a software framework for processing large data sets in a distributed fashion over a several machines. The core idea behind MapReduce is *mapping* your data set into a collection of <key, value> pairs, and then *reducing* overall pairs with the same key. The overall concept is simple, but is actually quite expressive when you consider that:

1. Almost all data can be mapped into <key, value> pairs somehow, and

2. Your keys and values may be of any type: strings, integers, dummy types…and, of course, <key, value> pairs themselves.

The canonical MapReduce use case is counting word frequencies in a large text, but some other examples of what you can do in the MapReduce framework include:

- Distributed Sort
- Distributed search
- Web-link Graph Traversal
- Machine Learning

In the past few decades, parallel, distributed and grid techniques were applied to data mining. For parallel and distributed paradigms, database was divided into several segments, which were distributed to different computing nodes for data mining. By such a strategy, the global computational effort is shared. And the computing efficiency increases because the subtasks operate on distributed data sites concurrently [2]. Knowledge grid offers tools and techniques for distributed mining and extraction of knowledge from data repositories available on the grid [3].
Since data mining tasks become increasingly complex as data accumulating, research in the past few decades was focused on parallel and distributed mining techniques. In most of the research, database was divided into several segments, which were distributed to different computing nodes for data mining. By such a strategy, the global computational effort is shared. And the computing efficiency increases because the subtasks operate on

distributed data sites concurrently [3]. However, the computing nodes will exchange transaction information among each other during the mining process. The high efficiency will be undermined by frequent and massive data interchanging. Meanwhile, information processing in network requires real-time communication. But parallel and distributed data mining do not guarantee excellent mechanism of information sharing and cooperation to fulfill such a significant requirement. In addition, the data privacy and security is also a major concern, since data may be illegally attacked when the parallel and distributed algorithms duplicates the database to every node [4].

Hence in this paper we use map reduce technique to determine classification rule mining model. The procedure of classification is arranged considering the distributed and parallel cloud environment. And the adapted jrip algorithm, which makes good use of the computing power of cloud computing, is designed to solve this model. For training and testing the proposed model, we use data collected from UCI dataset to conduct an illustrative example. Section 2 is a brief review of the literature relevant to data mining and map reduce workflow. Section 3 describes the time and accuracy of the model against different amount of data. Experiment in section 4 evaluate the validity and performance of the proposed model; Section 5 concludes the whole research.

II. RELATED WORKS

Distinguishing with the traditional mining paradigms, data mining in cloud is a novel area filled with valuable issues worthy of investigation. Basing on an intensive review on the relevant literature, most of the researchers concentrate on the following problems.

A. Data Mining Algorithm

Cloud computing, with its promise of virtually infinite computing and storage resources, is suitable to solve resource greedy computing problems. One problem of data mining in the cloud has been investigated from the data mining algorithm perspective. Wang et al. [9] utilized the powerful and huge capacity of cloud computing into data mining and machine learning. In their experiments, three algorithms, i.e., global effect (GE), K-nearest neighbor (KNN) and restricted Boltzmann machine (RBM) were performed in cloud computing platforms, which use the S3 and EC2 of Amazon Web Services. And they built two predictors based on KNN model and RBM model respectively with the order to testify their performance based on cloud computing platforms.

The MapReduce programming model was designed for processing massive data sets in a parallel network. Based on this programming model, Wang et al. [9] adapted the SPRINT algorithm which is ideal tool for data classification. SPRINT has been designed to be easily parallelized. Due to the parallelism, the original SPRINT was modified to be implemented in Hadoop architecture. The algorithm divided datasets in vertical direction and horizontal direction respectively, in accordance with the "Map" step in MapReduce. The vertical partition separated datasets by attribute, while horizontal partition produced many item sets. They applied the revised SPRINT algorithm to classify customer groups with different credit grades.

B. Architecture of Data Mining

Cloud is an infrastructure that provides resources and services over the Internet. Generally speaking, a cloud computing platform consists of a storage cloud, a data cloud and a compute cloud, which are responsible for storage services, data management and computational tasks. Google's App Engine platform is composed of Google File System (GFS), BigTable and MapReduce. Amazon provides its cloud services by Amazon Web Service (AWS), which contains Simple Storage Service (S3), Simple DB and Elastic Computing Service (EC2)].

Grossman et al. developed a cloud-based infrastructure to support data mining applications. The infrastructure consists of a storage cloud called Sector and a compute cloud called Sphere. The Sector storage cloud was designed for wide area, high performance networks and employed specialized protocols to utilize the available bandwidth, while GFS and Hadoop Distributed File System (HDFS) assume small bandwidth and don't work well with loosely coupled distributed environments. The Sphere compute cloud allows user defined functions.

C. A MapReduce Workflow

When we write a MapReduce workflow, we'll have to create 2 scripts: the map script, and the reduce script. The rest will be handled by the Amazon Elastic MapReduce (EMR) framework.

When we start a map/reduce workflow, the framework will *split* the input into segments, passing each segment to a different machine. Each machine then runs the *map script* on the portion of data attributed to it.The *map script* (which you write) takes some input data, and maps it to <key, value>pairs according to your specifications. . For example, if we wanted to count word frequencies in a text, we'd have <word, count> be our <key, value> pairs. Our map the data doesn't have to be large, but it is almost always much faster to process small data sets locally than on a MapReduce framework. Try downloading the Shakespeare corpus and running it through your

MapReduce scripts on your local machine script then would emit a <word, 1> pair for each word in the input stream. Note that the map script does no *aggregation* (i.e. actual counting) – this is what the reduce script it for. The purpose of the map script is to model the data into <key, value> pairs for the reducer to aggregate. Emitted <key, value> pairs are then "shuffled" (to use the terminology in the diagram below), which basically means that pairs with the same key are grouped and passed to a single machine, which will then run the *reduce script* over them2.The *reduce script* (which you also write) takes a collection of <key, value> pairs and "reduces" them according to the user-specified reduce script. In our word count example, we want to count the number of word occurrences so that we can get frequencies. Thus, we'd want our reduce script to simply sum the *values* of the collection of <key, value> pairs which have the same key.

The diagram below illustrates the described scenario nicely.

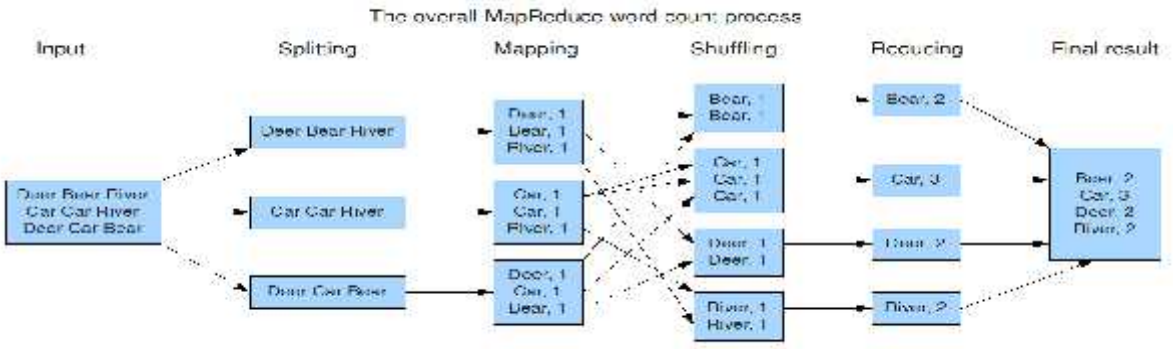

Fig 1: The overall MapReduce word count process

III. METHODOLOGIES

A. Classification

Classification is an important mission in data mining, and probably has become the most studied data mining task. In this task, the goal is to predict the value of a specified goal attribute (called the class attribute) based on the values of other attributes (called the predicting attributes) [14]. The dataset is symbolized by an attribute set consists of a number of attributes, R=(*a*1, *a*2,…, *a*N), where *ai*(*i*=1,2,…,N) is an attribute. The attribute set can be divided into two parts: 1) predicting attributes, C=(*c*1, *c*2,…, *c*m); 2) class attributes, D=(*d*1, *d*2,…, *d*n). A classification rule is demonstrated as:

If (*c*1∈*I*1)∧(*c*2∈*I*2)∧…∧(*cm*∈*Im*),

Then (*d*1∈*J*1)∧(*d*2∈*J*2)∧…∧(*dn*∈*Jn*)

Where I*i* and J*j* (*i*=1,…,m; *j*=1,…,n) are values of *ci* and *dj* respectively. "If" part contains a combination of conditions, and "Then" part contains the predicted class labels.

The data record is horizontally divided into two parts – training set and test set – that are mutually exclusive and exhaustive. The data mining algorithm discovers classification rules based on the training set, and evaluates the predictive performance of these rules with the test set.

The application of algorithm to a data set can be considered the core step of classification. The commonly used classification methods include: Bayesian reasoning, decision tree, genetic algorithm, rough set, case-based reasoning, neural network et al.

B. JRip Algorithm

This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP.

The algorithm is briefly described as follows:

Initialize RS = {}, and for each class from the less prevalent one to the more frequent one, DO:

1) *Building stage:*

Repeat 1.1 and 1.2 until the discretion length (DL) of the ruleset and examples is 64 bits greater than the smallest DL met so far, or there are no positive examples, or the error rate >= 50%.

1.1) *Grow phase:*

Grow one rule by greedily adding antecedents (or conditions) to the rule until the rule is perfect (i.e. 100% accurate). The procedure tries every possible value of each attribute and selects the condition with highest information gain: p(log(p/t)-log(P/T)).

1.2) *Prune phase:*

Incrementally prune each rule and allow the pruning of any final sequences of the antecedents.The pruning metric is (p-n)/(p+n) – but it's actually 2p/(p+n) -1, so in this implementation we simply use p/(p+n) (actually (p+1)/(p+n+2), thus if p+n is 0, it's 0.5).

2) *Optimization stage:*

after generating the initial ruleset {Ri}, generate and prune two variants of each rule Ri from randomized data using procedure 1.1 and 1.2. But one variant is generated from an empty rule while the other is generated by greedily adding antecedents to the original rule. Moreover, the pruning metric used here is (TP+TN)/(P+N).Then the smallest possible DL for each variant and the original rule is computed. The variant with the minimal DL is selected as the final representative of Ri in the ruleset.After all the rules in {Ri} have been examined and if there are still residual positives, more rules are generated based on the residual positives using building stage again.3.Delete the rules from the ruleset that would increase the DL of the whole ruleset if it were in it and add resultant ruleset to RS.

C. Classification Rules Mining in Cloud Computing Environment

1) Constraints analysis

Cloud computing is a network where distributed computer clusters constitute the resource pool of hardware. Tasks are divided into parallel segments, and assigned to available computing resources for processing. In that case, whether a computing task can be handled by cloud depends on decomposability and parallelism of the task. The requirements of processing computation tasks in cloud are defined as follow:

a) The task is capable of being divided into mutually exclusive sub-tasks.

b) Sub-tasks and data should be allocated to unoccupied processing nodes.

c) Mechanism of synchronization and communication amongst processing nodes is indispensable.

2) Model description

With storage and computing resources dispersing in the cloud environment, a Master is set as a central controller to search and distribute the resources. The responsibilities of Master are defined as: First, it divides the entitle task into several sub-tasks (here, a sub-task can be regarded as a data section), and assigns them to dispersed processing node; Afterwards, sub-tasks are executed on distributed nodes under the supervision of the Master; Eventually, the Master collects processing results on every node and synthesizes them to present a comprehensive set of classification rules as the mining result.
The entire procedure of classification rules mining consists of two stages: rules detection and classes induction. In the first stage, a Control Unit (CU) deals with a sub-task via translating each piece of data into a statement in form of “If … Then …” and detecting rules within the scale of sub-task. The results are raw rules. And in the second stage, Class Builders (CB) implement induction on the results from CUs. A CB induces rules with a certain class by analyzing the raw rules sharing this class attribute. Results of CBs are generalized as a rule set by Master and presented to users. Task distribution and coordination among CUs and among CBs are administered by the Master.
The classification rules mining model is displayed by Fig. 2 and the operating principle of it is explained explicitly as:

A mining task is requested to the Master. The Master searches available nodes which are not occupied by computation task, and set them as CUs. Then the Master divides the data file into data parts according to the quantity of CUs and the file size. If there are n nodes available and the data file size is measured as m, the size of every data part is defined as m/n.

Master sends a copy of a data segment to every CU.

CU translates every piece of data recording into a genetic chromosome according to the JRip encoding. Based on GA, every CU implements rules detections on the data segment it possesses. The detection results are stored in a

buffer and expressed in format of <key, value>, where key stands for class attributes and value stands for predicting attribute.

Master reads the data in buffer periodically and generates a list of <key, valueList> recordings, where key is the class attribute and valueList is a set of rules sharing the same class attribute.

When all the CUs finished their work, Master releases computing resource occupied by CUs.

Master searches available nodes to establish CBs, the quantity of which is the number of key value in the list of <key, valueList>. Then the Master distributes every CB with a piece of record in <key, valueList>, which is a set of rules sharing a certain class value.

On the basis of JRip, CB generates classification rules for a certain class value by implementing rules reduction on the rules set belongs to it, and transmits the results to the Master.

Master collects processing results from CBs and synthesizes them to present a comprehensive set of classification rules as the mining result.

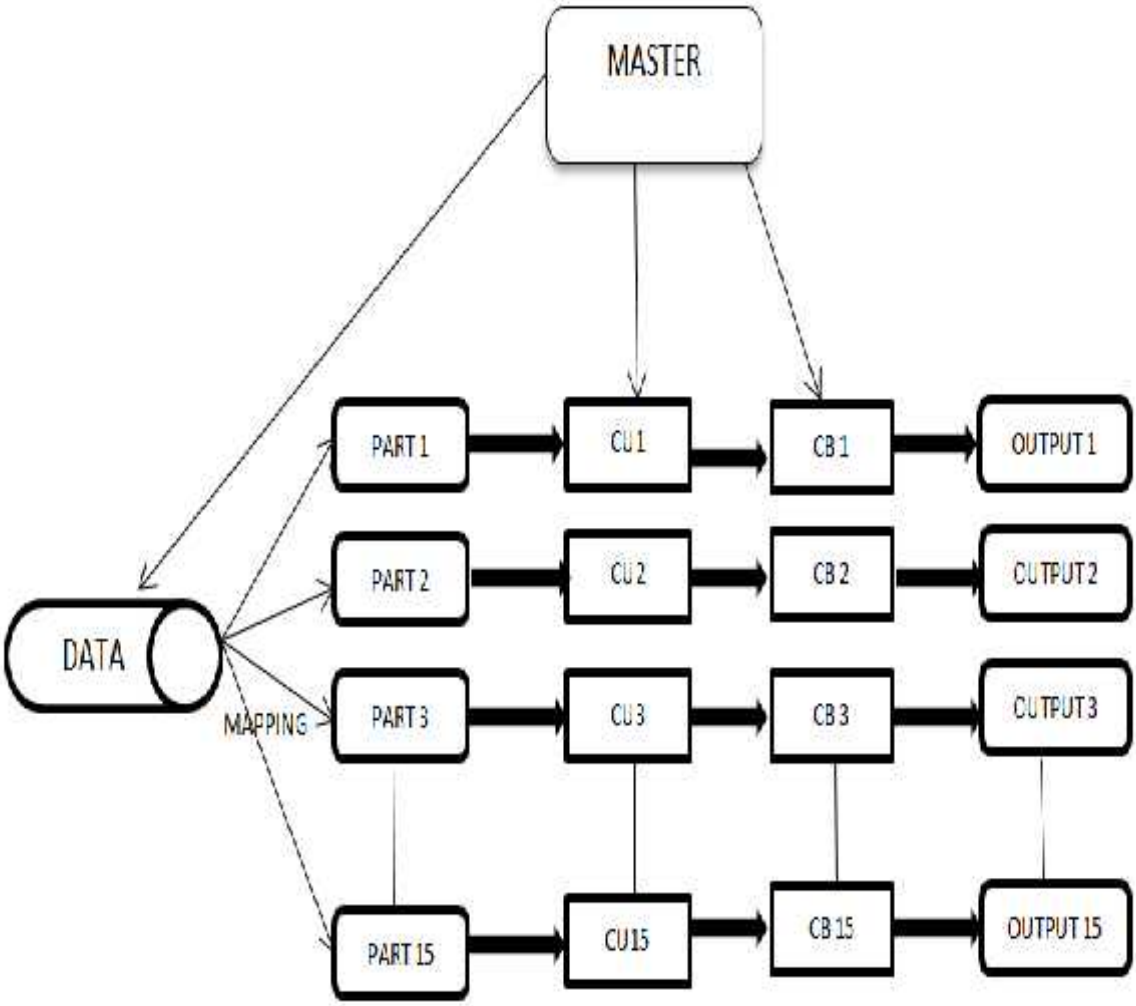

Fig 2: Classification rules mining model in cloud

The pseudo code of programming the model is displayed as:

partSet=Master.datapart(DateFile, 15);
partSet=(part(1), part(2),…, part(15));
For(i=1;i≤15;i++)
{CUData(i)=Master.copyData(part(i));
JRIPClassify(CUData(i));
//calculate classification rules in form of <key, value>
Input classy results into Buffer;}
For(j=1;j≤15;j++)
key(j).valuelist=Master.calculateValuelist(key(j));
//for each key in buffer, calculate a valueList consists of rules sharing this key.
For(j=1;j≤15;j++)
{CBData(j)=Master.distributeValuelist(ValueList(j));
JRIPClassify(CBData(j));
//calculate classification rules in form of <key, valueList>}
Master.getResults(CB);

IV. EXPERIMENT

In order to evaluate the validity of the proposed model, an illustrative example is applied to examine the performance. Data of breast cancer symptom, collected from the UCI dataset [16], are used as the raw data of our experiment. The dataset includes 700 pieces of data which are characterized by 11 mutually exclusive attributes. Two thirds of the data are defined as
the training set, and the rest is left for test set. Periods of the cancer corresponding to
different patients are classified with the proposed model based on their symptom. By comparing the result derived from the classification model with diagnosis concluded by the doctors, the validity of the proposed model could be assessed. So the classification of the above data set is done by the Weka tool.

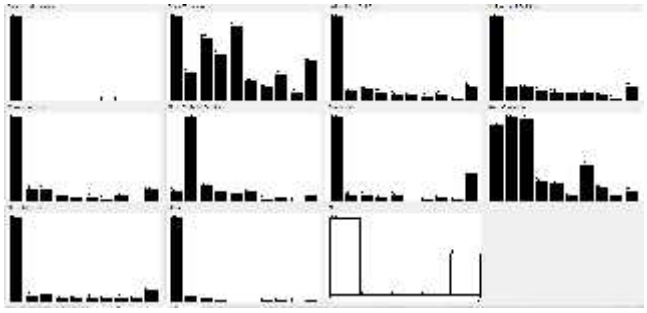

Fig 3: Discretized classification diagram of all the attributes of the dataset.

Classification accuracy and time consumption are two indexes indicating efficiency of the model.

So we plot the two graphs as accuracy and time consumption against the data(dividing them into segments).

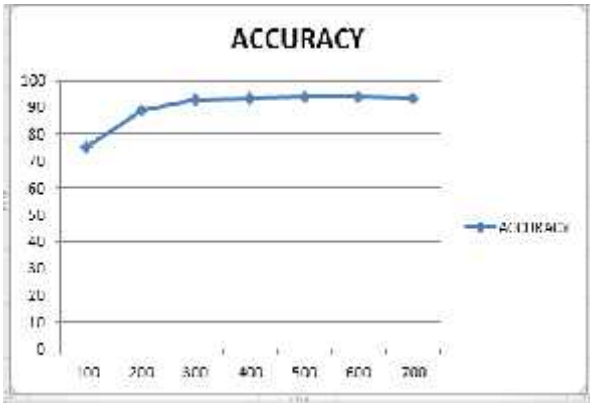

Fig 4:line graph of accuracy

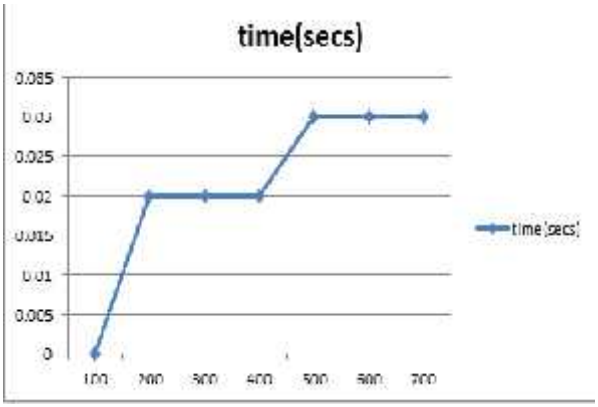

Fig 5:line graph of time

V. CONCLUSION

Map Reduce is a key technique for processing and generating a large amount of data which is used in data mining. So, we have developed solvent of classification rules mining model with jrip algorithm and map reduce technique. An illustrative example is analyzed to show the feasibility and effectiveness of the suggested model. The integration of classification and map reduce in data mining is in initial stage of research and needs further more improvement.

VI. ACKNOWLEDGEMENTS

The Authors would like to thank management of Bharath university, Chennai, India,for their support and encouragement for this research work

.

VII. REFERENCES

[1] Keahey, K., Figueiredo, R., Fortes J., Freeman, T., and Tsugawa, M. 2008. "Science Clouds: Early Experiences in Cloud Computing for Scientific Applications". In Proceeding of High Performance Computing and Communications.

[2] Park, B. H., and Kargupta, H. 2002. "Distributed data mining: algorithms, systems, and applications", Data Mining Handbook..

[3] Cannataro, M., Talia, D., and Trunfio, P. 2002. "Distributed data mining on the grid", Future Generation Computer Systems, 18, 1101-1112.

[4] Lin, K. W., and Luo Y. C. 2009. "A fast parallel algorithm for discovering frequent patterns". In Proceeding of IEEE International Conference on Granular Computing.

[5] Sakr, S., Liu, A., Batista, D. M., and Alomari, M. 2011."A survey of large scale data management appproaches in cloud environments", IEEE Communications Surveys & Tutorials, 13(3), 311-336.

[6] Agrawal, D., Abbadi, A. E., Antony, S., and Das, S. 2010. "Data management challenges in cloud computing infrastructures", Databases in Network Information Systems, 5999, 1-10.

[7] Pandey, S., Voorsluys, W., Niu, S., Khandoker, A., and Buyya, R. 2012. "An autonomic cloud environment for hosting ECG data analysis services", Future Generation Computer Systems, 28(1), 147-154.

[8] Wang, J., Wan, J., Liu, Z., and Wang, P. 2010. "Data mining of mass storage based on cloud computing". In Proceeding of 2010 Ninth International Conference on Grid and Cloud Computing.

[9] Wang, E., and Li, M. 2009. "Research on mass data mining under cloud computing", Modern Computer, 22-25.

[10] Bedra, A. 2010. "Getting started with Google App Engine and Clojure", Internet Computing, 14, 85-88.

[11] Wang, L., Tao, J., Kunze, M., Castellanos, A. C., Kramer, D., and Karl, W. 2008. "Scientific cloud computing: early definition and experience". In Proceeding of the 10th IEEE International Conference on High Performance Computing and Communications.

[12] Grossman, R. L., Gu, Y., Sabala, M., and Zhang, W. 2009. "Compute and storage clouds using wide area high performance network", Future Generation Computer Systems, 25, 179-183.

[13] Freitas, A. A. 2002. "A survey of evolutionary algorithms for data mining and knowledge discovery", Advances in Evolutionary Computation, 819-845