Implementation of neotype simple Bayesian algorithm by mapreduce and the application of discreteness and continuity in data mining

Xiliang Yan[*]

Zhengzhou University of Industrial Technology, Zhengzhou, China

Received 1 October 2014, www.cmnt.lv

Abstract

MapReduce is a programming model that can run in a heterogeneous environment. Its programming is simple and used for the parallel arithmetic of large-scale data sets. We do not need worry about the underlying implementation details. MapReduce is applied into the three arithmetic of data mining: simple Bayesian algorithm, K-modes clustering algorithm and ECLAT frequent item set mining algorithm. This paper put forward an improved simple Bayesian algorithm which was implemented by MapReduce based on MapReduce programming model and the existing research. It could deal with the application of data mining which both with the nature of discreteness and continuity. At the same time, combined with the ideas of each algorithm and the running mechanism of MapReduce, this paper put forward K-modes clustering algorithm and ECLAT frequent item set mining algorithm which was implemented by MapReduce. These implementations expanded the application range of the two algorithms from stand-alone to cloud computing platform. When facing huge amounts of data, it can effectively improve the work efficiency of the algorithm.

Keywords: MapReduce, simple Bayesian algorithm, data mining, cloud computing, work efficiency

1 Introduction

With the gradual development of the Internet, large amounts of data are continuously produced all the time. When dealing with huge amounts of data, the traditional data mining work will show up the shortcomings of small storage capacity, poor stability, time-consuming, etc. The put forward of cloud computing readily solved these problems. Cloud computing are a kind of internet-based computing and the further development of distributed computing, parallel processing and grid computing based on internet computing. For its low requirement for hardware, cloud computing can run stably in a heterogeneous environment established by the cheap commercial computer. However, it can show the excellent performance and its efficiency can reach the speed of stand-alone computer by dozens of times. The vigorous development of the cloud computing industry opened up a new path in data mining field.

At present, the data mining work based on cloud computing platform has made many achievements. Based on the design and implementation of data mining system [1], Zhang Kezhi of University of Electronic Science and Technology adopted web mining and discussed the characteristics of web mining and research of each functional module design. It mainly focused on the introduction of association rule, clustering algorithm and regression algorithm, which constructed web algorithm. Li Mingjiang, et al in data mining technology and its application [2] put forward that the needs of the social market could not be solved by traditional data query statistics. Data mining could meet the implementation of huge amounts of data information transferred to useful data warehouse, what's more, provided decision support for the development of all walks of life. Based on the implementation of cloud computing programming model by using MapReduce, various data mining algorithms are implemented. Based on the design and implementation of data mining platform realized by MapReduce [3], Huang Bin, et al proposed a data mining platform design based on MapReduce. As the huge-scale data computing platform, this design idea provided reference and compensation for insufficient in data mining, data visualization and business intelligence application. At the same time, huge-scale data mining was implemented based on this method. Based on the research of cloud computing data mining summary [4], Guan Wenbo, et al analyzed the current data mining problems aiming at the burgeoning cloud computing technology, and the advantages of cloud computing as well as the construction of data mining application platform of cloud computing.

2 Data processing model based on cloud computing

2.1 THE STRUCTURE AND OPERATION MECHANISM OF CLOUD COMPUTING

MapReduce is the programming model used for distributed computing of huge data sets which is stored in distributed file system. It adopts the idea of "divide and

[*] ***Corresponding author's*** e-mail: yanxiliangyxl@163.com

rule”, that is, to decompose large-scale data into many small-scale data, and to complete together by distributing them to multiple nodes in the cluster. This can effectively reduces the computation complexity, thus to achieve the goal of improving the operation efficiency. In order to carry out the task of MapReduce in cluster, the collaboration of a few parts is needed [5], as shown in Table 1.

TABLE 1 MapReduce cluster structural table

Structural elements	**Role**
The client	The interface for users interacting with the cluster
JobTracker	Be responsible for dispatch the implementation of entire works
TaskTracker	The real work performer
Distributed file system	(HDFS) used for the storage of input/output of data(HDFS)

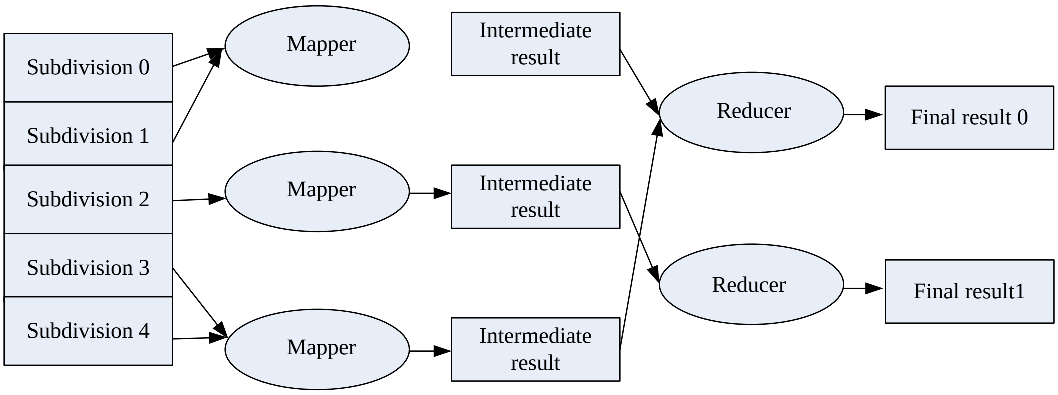


FIGURE 1 MapReduce operation mechanism

MapReduce is a simple framework [6]. Programmers only need to realize the Map function and reduce function. However, other issues, such as: distributed storage, job scheduling, load balancing, etc, which are all in the charge of MapReduce. Programmers do not need to notice that.

2.2 MAPREDUCE APPLICATION

Only by satisfying scalability, can tasks be performed in cluster [7]. The so called “scalability” task means the task that can be convergent and divergent with equal proportion, that is, the working method do not change after the convergent or divergent of certain task, and each node does the same work, such as number statistic, variance calculation, etc.

MapReduce task is of “divide first and combine second”. In the face of an extensible task, the task that can be distributed carried out is needed to be found out first, which is served as the core of Map function. For example, to look for a maximum of 100 million records, no matter how many data fragmentation it is divided into, each Mapper only need to do the same thing: to find out the maximum of current subdivision. If global variables is needed when carrying out task, the function provided by Hadoop can be used to set up that, and Distributed-Cache class can also be used to distribute the data file to each Mapper, thus to the overall synchronization of homework.

3 Massive data mining based on MapReduce

There are three typical algorithms in data mining: classification and prediction, clustering analysis and frequent item sets mining [8]. This paper took each representative type and used MapReduce programming model to realize it.

3.1 IMPLEMENTATION OF NEOTYPE SIMLE BAYESIAN ALGORITHM BY MPREDUCE

Simple Bayesian classification algorithm is a commonly used classification algorithm. It through calculate the prior probability of certain object to choose classes with maximum a posteriori probability as the objects belonging to the class [9]. That is Bayesian formula is used to calculate its posteriori probability, which is the probability of the object belongs to a certain kind.

Suppose there were *L* samples, and each sample is with *N* properties. There were almost *M* classifications $C_1, C_2, \ldots, C_{\mathrm{m}}$, and each sample could be expressed by attribute vector $X = \{X_{1,}, X_2, \ldots, X_{\mathrm{m}}, C\}$ of *N*+1 dimension. One unclassified sample *X* was used to predict the attribution that belongs to class C_i, if and only if:

$$P(C_{\mathrm{i}}|X) > P(C_{\mathrm{j}}|X), 1 \le \mathrm{j} \le M, \mathrm{j} \ne \mathrm{i}\,. \tag{1}$$

According to Bayesian equation, only if we obtained $P(X|C_{\mathrm{i}})$, can we got $P(C_{\mathrm{i}}|X)$. However $P(X|C_{\mathrm{i}}) = P(\mathrm{x}_1|C_{\mathrm{i}}) \times P(\mathrm{x}_2|C_{\mathrm{i}}) \times \ldots \times P(\mathrm{x}_{\mathrm{n}}|C_{\mathrm{i}})$, thus we needed to calculate the values from $P(\mathrm{x}_1|C_{\mathrm{i}})$ to $P(\mathrm{x}_{\mathrm{n}}|C_{\mathrm{i}})$ in training. Suppose that we calculated $P(\mathrm{x}_{\mathrm{k}}|C_{\mathrm{i}})$: if A was of discrete attribution, then $P(\mathrm{x}_{\mathrm{k}}|C_{\mathrm{i}})$ was the attribution of A_k took x_k, which can be summarized as the tuple number in C_i class divided the total number of that in sample *L*. If A_k was of continuous attribution, supposed it followed the average μ, the gauss distribution of standard deviation σ:

$$g\left(x_k, \mu_{C_i}, \sigma_{C_i}\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} . \tag{2}$$

Then $P\left(x_k \middle| C_i\right) = g\left(x_k, \mu_{C_i}, \sigma_{C_i}\right)$, that is, we needed to calculate the average and standard deviation of A_k in all the samples of C_i class.

There were two phases in the implementation of improved MapReduce: training and testing. In training phase, the statistical work of discrete attribute value and calculate work of the average and standard deviation of continuous attribute was carried out step by step. Therefore it was written into the Map function. After statistics, the statistical results were integrated and outputted according to requirement, and this should be completed by Reduce function. However, the result could not be used as classifier, and it also needed to be transformed. It can not be completed by Reduce function. And this program needed to be performed by the main function.

In testing phase, Map is responsible for data test, while Reduce conducts the final test of the accuracy of statistics. The process of algorithm learning by Simple Bayesian classifier of MapReduce was: to input training data set, to output classifier.

1) Map function Input was got from the distributed file system. In MapReduce framework, the key got from input file was the id of a certain sample and value was the content of that sample. For each sample *I*, in data *m*; for each property a, in each sample *i*; if (a was discrete), then *i* was the number of current value in this category. Else if (*a* was continuous), then statistic the sum and quadratic sum of this property value. After traversed the current subdivision, outputted the above statistics result.

2) Reduce function: all the temporary results from Mapper were read. The statistics results of discrete type were collected to determine the corresponding probability. In continuous data, supposed the sum of certain property value was $\sum X$, the quadratic sum was $\sum X^2$, the total sum of training sample was Sum:

$$\mu = \frac{\sum X}{Sum} \tag{3}$$

$$\sigma = \sqrt{\frac{\sum X^2}{Sum} - \left(\frac{\sum X}{Sum}\right)^2} \tag{4}$$

Then we got the average value and quadratic value if this property. After statistic of all the items, we got the final results of MapReduce task, and outputted it at the same time.

3) Main function Read the Reduce results. If the corresponding record of a certain <key, lue> key value was of disperse property, then it was transferred into the form of property mark, value, tag number, probability and outputted them. If the corresponding record of a certain <key, value> key value was of continuous property, then it was transferred into the form of property mark, value, average value, standard deviation and outputted them. Then the construction of classifier was finished.

3.2 IMPLEMENTATION OF K-MODES AGORITHM BY MAPREDUCE

K-modes algorithm is a kind of clustering algorithm, and it needs to looping perform. It calculates the difference of value, mode and the distribution execution of work of objective function. The mission of Reducer is to collect these values and generate a new cluster center. Main function also is responsible for judging whether the value of objective function change or not. If the value has not changed after two rounds of computation, then the mission is stopped. The process of clustering algorithm of dispersed K-modes is to input data sets and initiate clustering center, then to output the final clustering center.

1) Map function the initiated clustering centre was distributed to each Map node through DistributedCache. or each sample i, in data sets m; the algorithm defined formula was employed:

$$d(X,Y) = \sum_{j=1}^{m} \delta\left(x_j, x_j\right) \tag{5}$$

The difference between each *i* and clustering center was calculated, the tab of clustering center with minimum differences was used to mark *i*. The Equation (5) was used to get the objective function based on the differences between the two samples:

$$P(W,Q) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,\ l}\,\delta\left(x_{i,\ j}, q_{l,\ j}\right), \tag{6}$$

or each property a, in sample i; the mode of each property was collected and wrote in mode matrix. The mode matrix and objective function was outputted.

2) Reduce function all the temporary results were read from Mapper. The same position of each temporary mode matrix was added together and synthesized into a mode matrix, the value of objective function was collected. According to mode matrix, a new clustering center was obtained and updated. The value of objective function was passed to the main function.

3) Main function if the value of current objective function was different from the previous round calculation, then run the procedure in cycle, otherwise shuttled down the procedure. If one mission was with convergence properties, then this strategy could achieve a good effect.

3.3 IMPLEMENTATION OF ECLAT AGORITHM BY MAPREDUCE

ECLAT algorithm is a kind of method mining frequent patterns. For item set needs to be frequently search, many MapReduce mission also needs to be conducted. The output of the previous mission is served as the input of later mission, at the same time, the work of algorithm was carried out in distribution. Mining algorithms of Distributed ECLAT frequent item sets was to input data sets and support degree, output frequent item sets.

Task one:

1) Map function for each item in sample j was converted into the format of <item, TID_set>, for instance: <itema TID1, TID2… TIDM>. The item-sets one was outputted.

2) Reduce function the temporary files of each Map were integrated into vertical one item-sets <itema, TID1, TID2… TIDN>.

Task two: the variable K was introduced, and the initial value of K was supposed as 2.

1) Map function the frequent one item-set was obtained according to the first task. The vertical K item-sets was got based on the property of Apriori and was outputted.

2) Reduce function the temporary files of each Map were integrated into vertical K item-sets <itema, itemb, TID1, TID2… TIDW>, and judged whether it was frequent item sets. If it was, then outputted it, frequent K item sets was obtained. K=K+1. From task one to task N: the procedure was repeated in the task two and was stopped until the obtained frequent item sets was empty.

4 Experimental results

Given that simple Bayesian classification algorithm, K-modes clustering algorithm and ECLAT frequent item set mining algorithm studied in this paper were already the mature data mining algorithms [10], thus this paper only investigated the time efficiency and situation changes of each algorithm under the environment of cloud computing. Data set is the large data sets that artificially generated and each algorithm generated three orders if magnitude. The parallel computing was started from one compute node, increased one by one until it reached to 16 nodes. The time and speed-up ratio curve was drew according to the running condition, at the same time, the corresponding standalone version algorithm was applied to compare with that [11].

4.1 EXPERIMENTAL RESULTS OF THE NEOTYPE SIMLE BAYESIAN ALGORITHM

A single sample was with 10 discrete attributes, 10 continuous attributes, and one classification property. Continuous attributes was accord with gaussian distribution. 106 (the size of 54 MB), 107 (the size of 540 MB) and108 (size 5400 MB) samples were generated respectively as the training sample. The experimental results of the neotype simple Bayesian algorithm were shown in Figure 2. From Figure 2, we can see the parallel running time was much larger than single machine running time. This was due to:

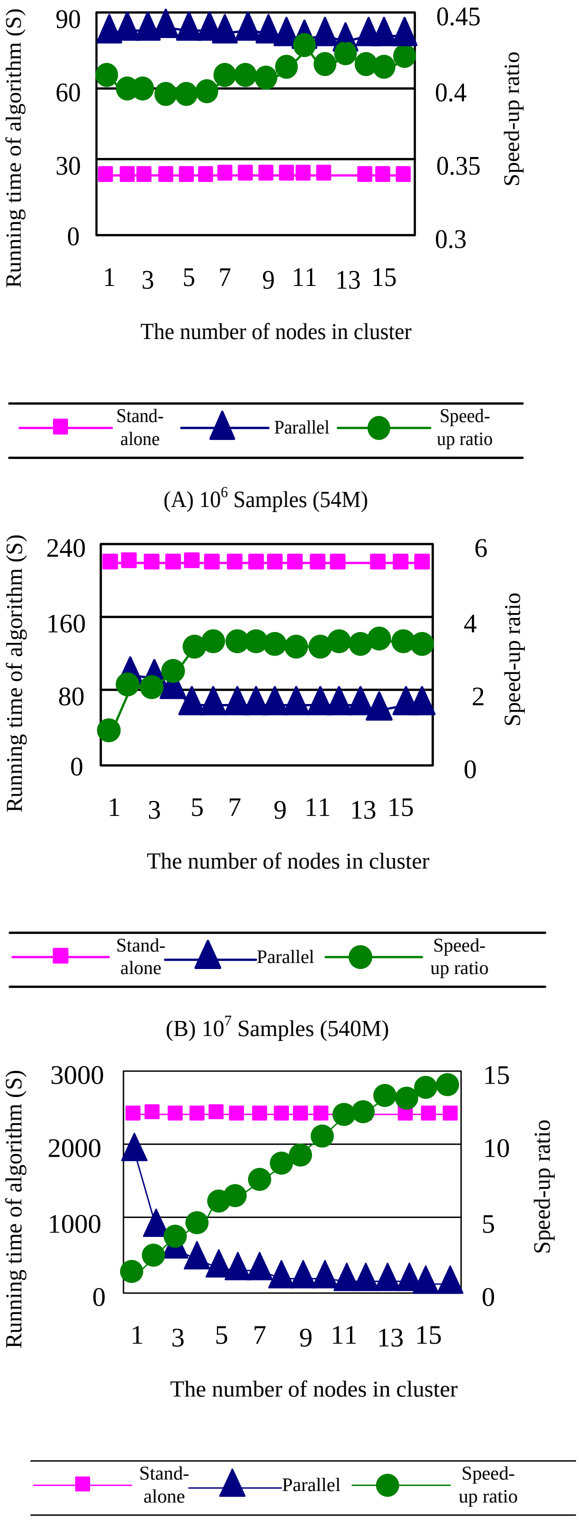
1) In default situation, MapReduce class library divided the file into the subdivision of 64 MB size. And this file was smaller and finally assigned to a Mapper to perform that. For one node could operated two Mapper task, it was finally assigned to one node to perform no matter how many nodes were in cluster.

2) The initialization of MapReduce task took a certain amount of time, small data set ran by stand-alone program was relatively flexible, thus stand-alone program took less time than parallel program. It was concluded that small-scale data set was not suitable for parallel computing. Figure 2B shown after the data become larger, the advantages of f cloud computing appeared slowly. Data of 540 MB would divide into 9 subdivisions. It was got from the theoretical analysis that 9 subdivisions needed to be assigned to five nodes to calculate. When the node number was less than 5, the running time would reduce with the increase of number of nodes. After the node number was more than 5, then the calculate nodes would meet the task. When the number of nodes continuous increased, the running time would not reduce for they had no effect on the task. We could see from Figure 2C, when the data was large enough, the running time would reduce continuous with the increase of the number of nodes in cluster. It was calculated that the running time of the task would reach to its minimum when the number of nodes was 42.

4.2 EXPERIMENTAL RESULTS OF K-MODES AND ECLAT

The experimental results of single sample of K-modes were 10 integers, and samples of 106 (size 21 MB), 107(size 210 MB) and 108(size 2100 MB) were generated respectively as the data sets. The experimental results of K-modes algorithm was shown in Figure 3.

ECLAT algorithm was quite special, for there was the operation of join and meet in each iteration operation which needed the combination of the two. If there were m samples, then the first join and meet operation needed to conduct for C_{m}^{2} times. The problem size changed to square times of previous one, so it was more quickly in small data. However, if the data size was quite large, then the stand-alone was restricted by internal storage and the calculation could not operate correctly. This experiment reduced the size on the data selection, and selected the sample of 103(size 32 kB), 104 (size 344 kB) and 105(3550 kB). At the same time, in order to made full use of the nodes in cluster, the default setting of Hadoop was changed. The default subdivision was changed from 64MB to 2MB. Each sample was of 1 to 20 unequal tuples. Through the experiment, these three experiments would all appeared the problem of memory overflow when conducted stand-alone calculation, then we drew that the experiments were only conducted based on cluster. The experimental results of ECLAT algorithm was shown in Figure 4

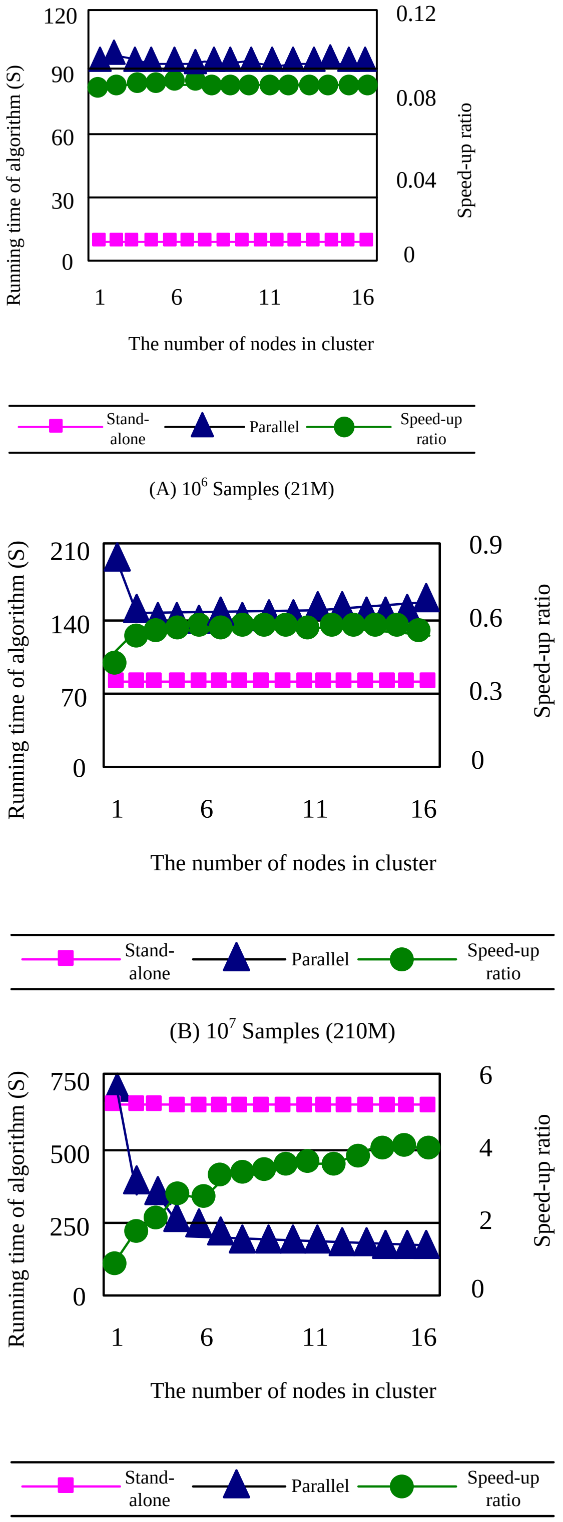


(A) 10^6 Samples (54M)

(B) 10^7 Samples (540M)

(C) 10^8 Samples (5400M)

FIGURE 2 The simple Bayesian training and speed-up ratio time of samples with different orders of magnitude

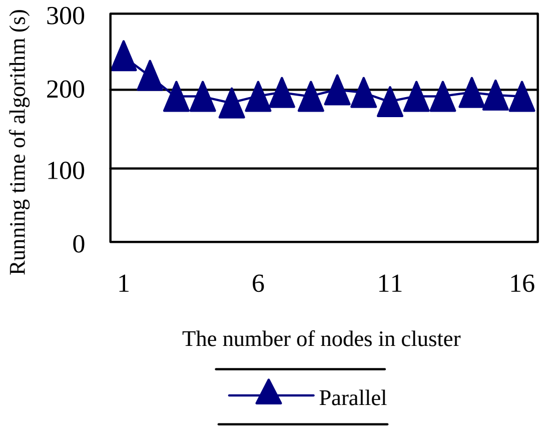


(A) 10^6 Samples (21M)

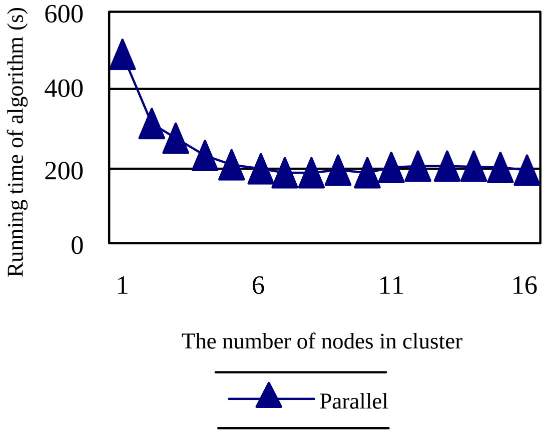
(B) 10^7 Samples (210M)

(C) 10^8 Samples (2100M)

FIGURE 3 K-modes clustering and speed-up ratio of samples with different orders of magnitude



(a) 10^3samples(32K)



(b) 10^4samples(344K)

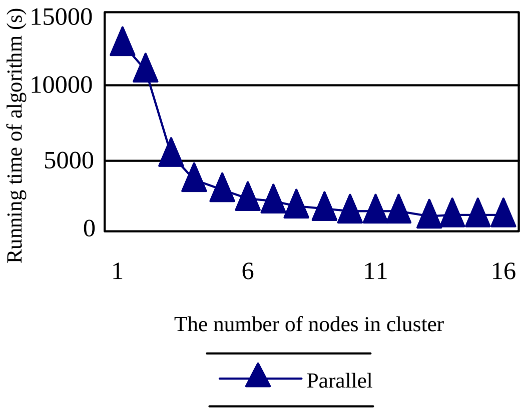


FIGURE 4 Figure of ECLAT frequent item set mining time of samples with different orders of magnitude

5 Conclusion

It was concluded from Bayesian experimental results that when the data was too small, the adoption of cloud computing could not reach the effect of acceleration. However, when the data was big enough and all the nodes in cluster conducted work, then the whole cluster can achieve its maximum efficiency. Generally, the time curve of K-modes algorithm was similar to Bayesian algorithm. The cycle index conducted in this algorithm was greatly related to the sample themselves and the selection of initial center class. It is still needed to pay attention to about how to select the initial center, reduce the cycle index and improve the efficiency of algorithm in distributed computation in future. The running time of ECLAT algorithm is composed of the spend time of each iteration. The running time was almost in line with rule mentioned above. This experiment also indicates that the subdivision size should be adjusted appropriately according to the data size, which is a way to improve the efficiency of distributed computing.

References

[1] Zhang K 2013 The Design and Implementation of Data Mining System Based on Web *Sichuan: University of Electronic Science and Technology*

[2] Li M J, Tang Y, Zhou L J 2012 Data Mining Techniques and Its Applications *China New Telecommunications* (22) 66-8

[3] Huang B, Xu S R, Pu W 2013 The Design and Implementation of Data Mining Platform Based on MapReduce *Computer Engineering and Design* **34**(2) 495-501

[4] Guan W B, Lei L 2013 The Research of the Summary of Data Mining Based on Cloud Computing *Horizon of Science and Technology* (33) 208-9

[5] White T 2010 Hadoop: The Definitive Guide. [S.l.]: Yahoo Press.

[6] Dean J, Ghemawat S 2008 MapReduce: Simplified Data Processing on Large Clusters *Communications of the ACM* **51**(1) 107-3

[7] Owen S, Anil R, Dunning T, Freidman E 2011 Mahout in Action [S.l.] Manning Publications

[8] Han J, Kamber M, Pei J 2011 Data Mining: Concepts and Techniques. [S.l.]: Morgan Kaufmann.

[9] Chu C T, Kim S K, Lin Y A, Yu Y Y, Bradski G, Ng A Y, Olunkotun K 2007 Map-reduce for Machine Learning on Multicore *Advances in Neural Information Processing Systems* 19

[10] Zhang C, Guo Y 2011 Data Mining and Cloud Computing—an interview with Dr. He Qing of Computing Technology Institution of the Chinese Academy of Sciences *Digital Communication* **38**(3) 5-7

[11] Zhang J X, Gu Z M, Zheng C 2010 Review of Cloud Computing Research Progress *Research of Computer Application* **27**(12) 429-433

Authors

Xiliang Yan, born 1968, Henan Province, China

Current position, grades: lecturer, engineer.
University studies: bachelor's degree in computer science, Beijing Normal University in 1990.
Scientific interests: CAT(computer application technology).