International Journal of Grid Distribution Computing
Vol. 8, No.3, (2015), pp.43-52
http://dx.doi.org/10.14257/ijgdc.2015.8.3.05

# An Improved Classification Course Based on Mapreduce

Haitao Wang[1,2], Shufeng Liu[1] and Zongpu Jia[2]

[1,2] *School of Computer Science and Technology*
*Jilin University*
*QianJin Street, ChangChun, JiLin, China*
*Jz_wht@126.com*
*Henan Polytechnic University*
*Shiji Street, Jiaozuo, Henan, China*
*Jz_wht@hpu.edu.cn*

***Abstract***

*It is an importance step for near-duplication detection to perform file classification in the data mining field, in this paper an improved classification course is proposed which consists of training and test course corresponding to its algorithm respectively. It utilizes the MapReduce computing model created by Google to conduct the classification calculation. Specially, the Sogou news data with various data amounts which simulated the massive data set was used for testing effectiveness and a comparative evaluation on execution time and speedup was accomplished on the experimental circumstance. The results obtained shows that the classification course obviously reduces the execution times greatly and gains the ideal speedup ratio when increasing data amounts, achieves the better performance.*

***Keywords:*** *Classification, Naïve Byes, Algorithm, MapReduce, Massive Data*

## 1. Introduction

With the advance of science and technology and development of society, there is a large amount of data which created constantly from various fields, such as SNS (social networking service), e-business, blogs and research institution and so on. Facing the quick growth of data volume [1], it's becoming more and more important to build the document classification system which is performed automatically and accurately and conducts massive files intelligent analyzing and managing [2].

There exists the extensive utilization about file classification technology. At present, some fields have achieved a great success, *e. g*, document filtering and indexing, semantic discriminating [3], and documents grouping and so on. While, influenced on some factor, such as computing and storage ability limitation, the traditional classification technology expose some defects that is the poor stability, long running time and lack of storage space when handling the massive data. Clouding computing creates an ideal chance to resolve the above problem, catering to these real requirements [4].

In complex classification domains, such as intrusion detection systems (IDSs), features or attributes may contain false correlations, which hinder the underlying process and in general, the learning task to be carried out. Furthermore, some features may be irrelevant and some others may be redundant since the information they add is contained in other features [5]. These extra features can increase computation time, and can have an impact on the accuracy of the classifier built. For this reason, these classification domains seem to be suitable for the application of feature selection methods. These methods are centered in obtaining a subset of features that adequately describe the problem at hand without degrading (and most of the times, improving) performance [6]. Feature selection is primarily performed to select relevant informative features, but it can have other

ISSN: 2005-4262 IJGDC

motivations, including general data reduction, feature set reduction, performance improvement and better data understanding. There are two main models dealing with feature selection: filter methods and wrapper methods (Kohavi and John, 1997). Wrapper models optimize a classifier as part of the selection process, while filter models rely on the general characteristics of the training data to select the best features with independence of any classifier. Wrapper methods use a classifier and a search technique to score subsets of features according to their predictive power. That is why they tend to obtain better performances than filters, even at the expense of a higher computational cost. On the other hand, when dealing with large datasets, filters are the most suitable alternative. Feature selection methods are adequate for both classification and regression tasks, although most research is done related to classification. In the feature selection field, it is difficult to find methods that can deal directly with multiple class problems, as very little research has been done in this aspect (Bruzzone and Serpico, 2000; Chiblovskii and Lecerf, 2008). In these studies, the multiple class approaches are shown to suffer from the so-called accumulative effect, which becomes more visible when the number of classes grows and results in removing relevant and un-redundant features.

The main difficulties to be taken into account in multiple class algorithms are the following:

- The dataset presents one or several classes that contain a considerable higher number of samples than the data of the other classes (unbalanced classes).
- Determining which features are appropriate for each class is complicated, because feature selection results in a set of attributes that could represent only the majority classes (Bosin, *et. al*., 2007).

To solve the above task which discussed, this paper presents an improved naïve Bayes classification algorithm [7][8] based on cloud computing-Mapreduce, and verifies the feasibility by means of experiments and the simulated data set. This paper is organized as follows: section II introduces the relevant work in the aspect of file classification, section III discusses the file classification course and presents the improved aspects of classification course happed in the two stages in details, an experiment is implemented and simulated on the classification course and the comparative outcome is shown in the section IV, section V summarizes the whole paper of work and points the research direction in the future.

## 2. Relevant Work

There are some classic classification algorithms in the field of data mining [9-10], *e. g.*, SVM (Support Vector Machine), Naïve Bayes model, k-nearest neighbor (KNN), decision tree and maximum entropy model, while, SVM, although being better overall performance than the other algorithms, has the stronger dependency on file sets about the sorting result [11]. With respect to the research on naïve Bayes model, Chow and Liu propose a tree form learning algorithm, Cooper and Herskovits present a k2 algorithm based on evaluation function and upper searching policy, Remco, at the basic of k2 algorithm, gives a novel k3 algorithm by utilizing MDL to substitute evaluation function, Friedman achieves parameter optimization of EM algorithm by means of combining naïve Bayes arguments learning with structure learning [12].

In the field of documents processing, the massive information resource provides an extensive application platform for algorithm. At present, the naïve Bayes algorithm has been applied on the file index, information extraction and grouping, for example, binary feature vector multiple Bernoulli naïve Bayes model, polynomial naïve Bayes model based on word frequency feature and so on. To break through the limitation of conditional independence, Friedman offers an enhanced tree form naïve Bayes model [13], which makes better the performance in the course of documents classification. Mieczyslaw, conducting a further improvement on the basic of enhanced tree form model [14], presents

 

a multi nets naïve Bayes model. Kononenko, using the relevant feature aggregation approach [15], comes up with a kind of semi naïve Bayes model to reduce negative effects on independence assumption. Daphne, aiming to the classification requirement of hierarchical type documents, puts forward a multilayer naïve Bayes model. Raina raises a kind of parameter learning methods based on maximum likelihood and maximum conditional likelihood. However, among above methods, tens of thousands of features seriously affects performance efficiency of running naïve Bayes algorithm.

At present, there are few research results of data mining based on cloud computing, *e. g.*,Apache Mathout project team puts forward a parallel data mining algorithm towards business field, the parallel distribute data mining platform-PDMiner is presented by computing institution of Chinese Academy of Sciences, which is able to perform the massive data process with TB data volume ranges, the BC-PDM is offered by China Mobile, which is a parallel data mining tool and provides the Web service model. All of these landmark achievement obtained push the research on classification algorithm forward greatly.

At the basic of Mapreduce programming model, there also little achievement on data mining, in 2007, CHU *etc*., proposed a kind of naïve Bayes algorithm based on Mapreduce, which principle is that constructing classifier is created on the circumstance of samples processed in the form of decentralized statistics and centralized integration. However, this method only can process the discrete type data and is unable to process the sequential type data.

Considering above task which have discussed, this paper puts forward an improved naïve Bayes classification algorithm, which can deal with the discrete and sequential type data and process massive data volume based on MapReduce programming model [16-17], furthermore, experiment proves that this measure designed is feasible and can obtain the high speed-up ratio and lower executing time[18].

## 3. Classification Course

The classification is the identifying course which determines a file belonging to the class set. From the math angle, the categorization is the course which maps the file without labeling category to the one with a certain category, and the course can be a one to one mapping, or a one to many mapping. It can be indicated by the math formula as follows: $f: A \rightarrow C$, where A is the file set which will be classified and C is a certain category set. Actually, the file classification is an application on pattern category [19], so many pattern algorithms can apply on the file classification. However, there are many different applications comparing to the other pattern category because the file classification is the inter-discipline between the pattern and the nature language.

- *High dimensional feature space* There are a large number of candidate feature when extracting the feature of file, even a training set which has the 1000 files, ten thousands of candidate sets will be produced in general.
- *Feature grammar correlation* Although mutual independence between features can avoid the matter of Gaussian disaster, theory research and much practice all show that an ideal file classifier should take into account the relationship between features of files, if not, it causes a great number of information loss.
- *Phenomenon of poly semantic word or synonyms* Whatever word, phrase or N-Gram being the feature of file, can't clearly express the semantic meaning in the special file, that is, the ploy semantic phenomenon, e.g. professor being a noun indicates a title of a technical post, teach being a verb indicates an action concept of lecturing knowledge. Some words with identical concept can be expressed by different feature, which is also the ploy semantic phenomenon, *e.g.* the two words, motor and car, show the same concept.
- *Sparse Feature* It' s in general very high for the dimension value of file feature

vector space, while those words being the file future is often normal frequency ones which emerge in the corpus. For a normal file, the appearance rate is zero for the most feature words in the vector space, which means that the most feature value is zero in file vector and the distribution of feature is very sparse.

According to the possible category in file, file classification falls into two kinds: that is, two categories and multi categories, wherein two categories of file classification is the simplest one. To the byes classification model, the solution approaches are similar for file classification of two categories and multi categories [20]. Therefore, it's unnecessary to conduct the special research on solution approach.

### *A. Discretization*

Many classification algorithms are shown to work on discrete data. In order to deal with the numeric attributes, a common practice for those algorithms is to discretize the data before conducting feature selection. For both users and experts, discrete features are easier to understand, use, and explain and discretization can make learning more accurate and faster. In general, the results obtained (decision trees and *etc*.) by using discrete features are usually more compact, shorter and more accurate than by using continuous ones, hence the results can be more closely examined, compared, used and reused. In addition to the many advantages of having discrete data over continuous one, a suite of classification learning algorithms can only deal with discrete data. Besides, the Weka tool (Witten and Frank, 2005) is used for the filtering process and it uses discretization by default [21]. The discretization method that it employs is Entropy Minimization Discretization. The final reason is that the dataset seems to be adequate to perform a discretization in some features. In essence, the process of discretization (Janssens *et al.*, 2006) involves the grouping of continuous values into a number of discrete intervals. However, the decision of which continuous values group together, how many intervals to generate, and thus where to position the interval cut points on the continuous scale of attribute values is not always identical for the different discretization methods. There exist many discretization approaches in the literature, but in this work the most suitable for large datasets have been chosen (Yang and Webb, 2002). All of them are described in the following subsections. Previously, some notation considerations are given: suppose a numeric attribute $X_i$ and n training instances for which the value of $X_i$ is known, the minimum and the maximum value are $f_{min}$ and $f_{max}$, respectively. All the discretization methods first sort the values into ascending order.

Assume that there are $L$ numbers of samples, which owns $N$ numbers of attributes, while the sample has M numbers of categorization in total, that is $C_1,C_2,...C_m$ , so, each of sample can be indicated by means of N+1 dimensions attribute vector $X=\{X_1, X_2, ..., X_m,\}$ , Given a sample x which is uncategorized, forecasting that X belongs to a special class $C_i$ equals the formula $P(C_i|X) > P(C_j|X)$ $(1\leq j\leq M, \ i\neq j)$.

According to the naïve Bayes formula, if computing $P(C_i|X)$, what the task to do is to obtain the value of $P(X|C_i)$, where $P(X|C_i)= P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$, so, just calculate the value from $P(x_1|C_i)$ to $(x_n|C_i)$, which the approach is list as follows:

*If* $A_k$ has the discrete attribute, $P(x_k|C_i)$ is the total number which the tuple number classified to $C_i$ class divides the L samples when the $A_k$ equals $x_k$.

*If* $A_k$ has the sequential attribute, suppose it obeys the Gaussian distribution that the mean value is μ and the standard deviation is σ, the calculation formula is defined as follows.

$$g(x_k, \mu_{c_i}, \sigma_{c_i}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The task of the Byes text categorization is to associate the text which will be classified, that is the $X(x_1, x_2,\ldots, x_n)$, with the most relative class $C(c_1, c_2,\ldots, c_j)$, where the $X(x_1,$

 

$x_2,\ldots, x_n)$ is the feature vector of text $X_q$ which will be classified, $C(c_1, c_2,\ldots, c_j)$ is the category system which is set. This means that the probability value, that is $P_1, P_2,..., P_n$ , is calculated that the vector $X(x_1, x_2,\ldots, x_n)$ belongs to the present class $c_1, c_2,\ldots, c_j$. where $P_j$ is the probability value of $X(x_1, x_2,\ldots, x_n)$ belonged to $C_j$, so the $max(P_1, P_2,..., P_n)$ corresponds to the category which the text belongs and the task of categorization turns into the maximum value of solution formula 1.

$$P(c_j \mid x_1, x_2, \cdots, x_n) = \frac{P(x_1, x_2, \cdots, x_n \mid c_j) P(c_j)}{P(c_1, c_2, \cdots, c_n)} \qquad (1)$$

Where the $P(c_j)$ stands for the probability of text belonging to the categorization $c_j$ , $P(x_1, x_2,\ldots, x_n \mid c_j)$ stands for the probability of the probability of text which will be classified belonging to the categorization $c_j$. $P(c_1, c_2,..., c_n)$ stands for the joint probability of all categories which presented. It's obvious fact that with regard to all the categories presented and the denominator is the constant number in formula 3, so the solution of the maximum in formula 1 turns into the formula 2.

$$c_{NB} = \arg\max_{c_j \in C} P(x_1, x_2, \cdots, x_n \mid c_j) P(c_j) \qquad (2)$$

According to the supposition of Byes, if the feature vector of text belongs to $x_1, x_2,\ldots, x_n$ independent and same distribution, the joint probability distribution equals to the accumulation which each probability distribution of attribute feature multiples mutually, that is $P(x_1, x_2, \cdots, x_n \mid c_j) = \prod_{i=1}^{n} P(x_i \mid c_j)$ (3)

Therefore, the formula.3 becomes the formula.4 as follows.

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(x_i \mid c_j) \qquad (4)$$

Namely, the solution of formula 4 is the classifying function and calculates the maximum value of function, the estimation of probability value distribute is the $\hat{P}(c_j) = \frac{M(C = c_j)}{N}$ , where $M(C= c_j)$ stands for the text number belongs to the $c_j$ category in the training text, $N$ for the total number of training text set.

$$\hat{P}(x_j \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + M}$$

Where $N(X_i = x_i, C= c_j)$ stands for the number of training text that the category $c_j$ belongs to the attribute $x_i$ , $N(C= c_j)$ for the number of training text, the $M$ for the number of key words of training text set which rejects the nonuse words and conducts preprocessing of text.

In the course of data mining, the data classification process consists of two stages, first, the training stage, which creates data class or predefines classifier of concept set, so, the classifier is constructed by means of analysis or learning from training set, then, the test stage, which obtains the categorization result by conducting classifier created categorizing. Therefore, the Byes classification algorithm on Mapreduce also includes the training course algorithm and the forecasting course one.

### *B. Training course*

Any training set should contain enough samples, as representative as possible, in order to get a good ‘‘model’’ and to enhance its generalization capabilities.

At this stage, count the discrete attribute value and calculate the mean value and

standard deviation in the distributed style, then, input the value into map function, after the computing course, output the conform result by the reduce function. However, the result can't be utilized by classifier and the transformation must be implemented still, this stage hands over the main function to do, the above course detailed by pseudo code is listed as follows.

***Algorithm 1. Training Stage***

*Input:* the training data set
*Output:* classifier
*Map function*:
***Step 1***. Obtain the input file from the distribute file system and get the id number of a sample.
For each sample (marked as *i*) in data set *(*marked as *m)*
For each attribute (marked as *a*) in each sample (marked as *i)*
if (a is the discrete type)
Count the number of current attribute value.
Else if (a is sequential type)
Calculate the sum and square sum of attribute value.
***Step 2.*** Output the above result after checking the current part.
*Reduce function*
***Step 1.*** Read all the temporary result from Mapper.
***Step 2.*** Gather all the discrete type result and obtain the corresponding probability.
***Step3.***With regard to the sequential data, obtain the mean value and standard deviation value.

$$\mu = \frac{\sum X}{n} \quad \cdots\cdots ① \qquad \sigma = \sqrt{\frac{\sum X^2}{n} - \left(\frac{\sum X}{n}\right)^2} \quad \cdots\cdots ②$$

Finishing all the process above, the result of Mapreduce task is obtained and output.
*Main function*
***Step 1.*** Obtain the result of reduce function.
***Step 2.*** If the value of <key, value> pairs records the discrete attribute, turns into <attribute, value, id, probability value> style and output.
If the value of <key, value> pairs records the sequential attribute, turns into <attribute, value, mean value, standard deviation value > style and output.

### *C. Test Course*

At this stage, map function is be in charge of test data and reduce function is responsible for precise rate test, which is list by pseudo code as follows.

***Algorithm 2. Test Course***

*Input:* the sample of test and classifier.
*Output:* the forecasting result.
*Map function:*
***Step 1.*** Distribute the file of classifier to every map node by distribute cache.
***Step 2.*** Read the files of classifier in the Mapper, if the recorder is the discrete attribute, read -in without any operation, if the sequential attribute, read in memory after calculating the corresponding probability of attribute.
***Step 3.*** Until the file information of classifier uploads completely, each of data which will be test is categorized according to classifier and the outcome is output.
*Reduce function:*
Gather all the result of categorization and output.

 

## 4. Experimental Test

As it was present in former section, several approaches exist to deal with problems that involve more than two classes. Depending on the classification technique chosen, different training datasets were constructed, it's important to remind that although several training sets were employed, the test dataset did not change and then it was the same used in the Sogou news data set.

### *A. Classification*

Now, utilizing the naive byes classifier resolve the classification task which determines whether or not you can play tennis according to the climate situation. Assume there are 14 training cases listed in table 1, where attributes of every day consist of outlook, temperature, humidity and wind, the attribute of class is the sport of playing tennis.

**Table 1. Training Case Prediction of Playing Tennis**

| Day | Outlook | Temperature | Humidity | Wind | Playing Tennis |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

For example, test a case X: <Outlook = sunny, Temperatue = cool, Humidty = high, Wind = strong> and determine this day suits whether or not playing tennis.

Obviously, our task is to forecast the class attributes value of new case which equals yes or no. So, the classifier of naïve byes is constructed shown in Figure3, where class node C indicates class attribute-playing tennis, other four nodes indicate the outlook, temperature, humidity and wind respectively, class node is the parent node of all attribute nodes, there are no relation of dependency.

According to formula 4, to obtain the value of c, it's necessary to calculate the probability of 14 training case from the table 1 and the calculation course is listed as follows:

$P(yes) = (9+1)/(14+2) = 10/16$
$P(sunny|yes) = (2+1)/(9+3) = 3/12$
$P(high|yes) = (3+1)/(9+2) = 3/11$
$P(no) = (5+1)/(14+2) = 6/16$
$P(cool|no) = (1+1)/(5+3) = 2/8$
$P(cool|yes) = (3+1)/(9+3) = 4/12$
$P(strong|yes) = (3+1)/(9+2) = 4/11$
$P(sunny|no) = (3+1)/(5+3) = 4/8$
$P(high|no) = (4+1)/(5+2) = 5/7$
$P(strong|no) = (3+1)/(5+2) = 4/7$

So, the value of formula 6 is: $P(yes) \times P(sunny|yes) \times P(cool|yes) \times P(high|yes) \times P(strong|yes) = 0.0069$

$P(no) \times P(sunny|no) \times P(cool|no) \times P(high|no) = 0.0191$.

From the above calculation, it's can be seen that the classification case is no according to the naïve byes classifier. Normalize the above possibility, the value that the naïve byes classifier categorizes the case-no equals 0.0191/(0.0191+ 0.0069) = 0.7346.

### *B. Experimental Comparison*

In the next steps, to assess the performance of classification course which improved, experiment is conducted on on the high performance computing platform of henan

polytechnic university, which utilizes the RedHat Linux to construct the cluster, every node including Intel Xeon E5504 2.00GHz CPU, 8G memory, 120Gdisk, J2SE 6.0/Hadoop 0.20.2, HBase-0.20.6 *etc*.

There are several metrics to the performance of classification methods. Here the execution time and speedup ratio is select as the assessing measure and classification course is calculated on Mapreduce computing model. The experimental data is collected from Sougou news data [22], which randomly selects 50M, 500M, 5000M data amount respectively as the training sample, a single sample owns the ten discrete attributes, ten sequential attribute (conformed to the Gaussian distribution) and one classifying attribute, the execution time and speedup of those samples is test under the single and distribute environment, the result which writes down is listed in Figure 1, Figure 2 and Figure 3 as. Follows

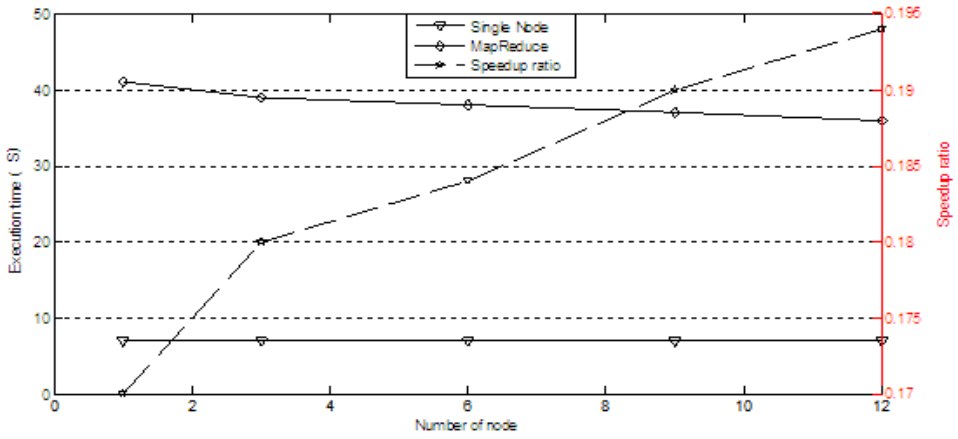


**Figure 1. 50M Sample Data Test**

From the Figure 1, it's an obvious fact that the execution time on single machine is far lower than one on the circumstance of parallel machines; there are some factors which resulted in this phenomenon as follows.

At the default circumstance, the file is divided into some blocks which size is 64M by the class library of Mapreduce. Because the 50M sample data is small relatively, a block is enough to divide and assigned to a mapper to run. Generally, a node can run two task of mapper, so, whatever how many nodes the cluster has, the 50M sample data only is allocated to a node to run. It costs a part of time on the task initializing of MapReduce, oppositely, it's flexible and quick for a single machine program to run. Therefore, it takes the little time on running the operation and it safely concludes that a scale amount of data don't suit for the parallel computing.

 

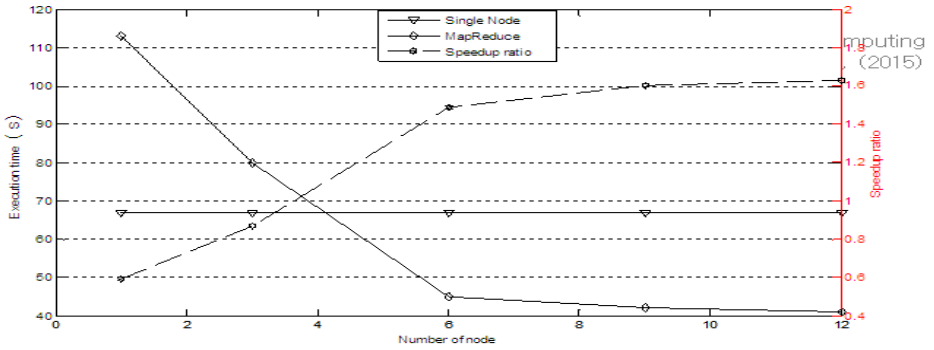


**Figure 2. 500M Sample Data Test**

The Figure 2 indicates that the advantages of MapReduce appear slowly when the amount of data becomes large. The reasons attributes to this facts that the amount of data which size is 500M can be divided into 9 blocks and is allocated 5 nodes to run, when the number of node is less than 5, the execution time reduces increasingly with the number of node increasing, while the execution time keeps be constant by and large when the number of node is more than 5, what takes place lies in the reason that the increasing of number of node does influence on the time of running when the node of computing satisfies the requirements of task.

The Figure 3 shows that the execution of time reduces constantly with the number of node in the clusters increasing when the data amount is big enough, until the number of node is 40, the time to run will reach the minimum.

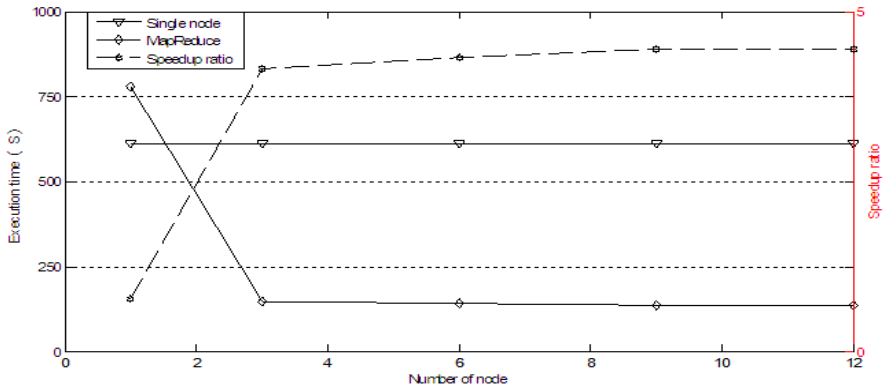


**Figure 3. 5000M Sample Data Test**

From the analysis what discussed above, it can be drawn a conclusion that when facing the task of classification, using the Mapreduce computing mode can't reach the goal of speedup under the circumstance of small data amount, the whole cluster can obtain the ideal efficiency when the data amount is big enough and every node of cluster runs normally.

## 5. Conclusions

This paper proposes an improved classification course based on Bayes classification, which consists of two stages, that is, first, construct the classifier by the means of parsing or learning from training sets for the data class predefined and concept set at the training stage, then, obtain the classification result by classifier which set at the test stage. The improved algorithm can suit the discrete and sequential attribute data classification and the

experiment course shows that this algorithm, when dealing with the massive data classification, can effectively reduce the complexity of task and the execution time, improve the running efficiency, which provides the scientific value for settling the massive data classification about the design idea in the future.

## Acknowledgment

This research is supported by the National Nature Science Foundation of China (No.61300216), the National Innovative Approach Special Project (No.2012IM010200), the Science and Technology Research Projects of HeNan Province (No. 132102210123), the Open Laboratory Project of Mine Information Key discipline of Henan University, the Doctor foundation of HPU (B2009-21)

## References

[1] J. McKnight, T. Asaro and B. Babineau, "Digital archiving: End-User survey and market forecast", **(2006)**, pp. 2006-2010. http://www.enterprisestrategygroup.com/ESGPublications/ReportDetail.asp?ReportID=591.
[2] D. P. Anderson, "BOINC, a system for public-resource computing and storage[C]//GRID2004", Proceedings of Fifth IEEE/ACM International Workshop on Grid Computing, Washington, DC:IEEE Computer Society, **(2004)**, pp. 4-10.
[3] A. Chowdhury, O. Frieder, D. Cnossman and M. C. McCabe, "Collection statistics for fast duplication document detection [J]", ACM Transaction on Information Systems, vol. 20, no. 2, **(2002)**, pp. 171-191.
[4] C. K. Chow, C. N. Liu," Approximating discrete probability distributions dependence trees", IEEE transactions on Information Theory, IT, vol. 14, no. 3, **(1998)**, pp. 462-468.
[5] E. Hemkovits, "Computer-based Probabilistic-Network Construction", USA, Stanford University, **(2001)**.
[6] R. A. Bouekaort, "A Stratified Simulation Scheme for Inference in Bayesian Belief Networks", In Uncertainty in AL Proceedings of the Tenth Conference, **(2004)**, pp. 110-117.
[7] N. Friedman, "The Bayesian structural EM algorithm", in Uncertainty in Artificial Intelligence (UAI). **(1998)**, pp. 129-138.
[8] L. M. D. Campos, J. M. Fermandez and J. F. Hucte, "Building Bayesian Network-based Information Retrieval Systems", In Proceedings of the 11 International Workshop on Database and Expert SystemsApplications，**(2000)**, pp. 543-550.
[9] I. Peshkin and A. Pfeffer, "Bayesian Information Extraction Network", IJCAL, **(2003)**, pp. 421-426.
[10] K. M. Chai, H. T. Ng and H. L. Chleu, "Bayesian Online Classifiers for Text Classification and Filtering", Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, **(2002)**, pp. 97-104.
[11] K. M. Ting and Z. Zheng, "Improving the performance of boosting for Naive Bayesian classification", Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, **(1999)**, pp. 296-305.
[12] K. Nigam, K. Andrew, M. Callum, S. Thrun, and T. M. Mitchell, "Text classification from labeled and unlabeled documents using EM", Machine Learning, vol. 39, no. 2-3, **(2000)**, pp. 103-134.
[13] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian network classification", Machine learning, vol. 29, **(1997)**, pp. 131-163.
[14] A. Mieczyslaw and Klopotek, "Very Large Bayesian Networks in Text Classification", International Conference on Computational Science, **(2003)**, pp. 397-406.
[15] I. Kononenko, "Semi-naive Bayesian Classifier. In Proceedings of the 6th European Working Session on Learning", **(2001)**, pp. 206-219.
[16] A. Hadoop, "Hadoop [EB/OL]. [2011- 03-15]. http:// hadoop. Apache.org".
[17] K. Zheng and Z Weimin, "Research situation of cloud computing system case", Journal of Software, vol. 20, no. 5, **(2009)**, pp. 1337-1348.
[18] L. Jianjiang, C. Jian, *etc*., "MapReduce Research review on parallel programming model [J]", Journal of electronics, vol. 39, no. 11, **(2011)**, pp. 2639-2645.
[19] D Jason, Lawrences, T. Jaime, *et al.*, "Tracking the poor assumptions of Bayes text classifiers [C]// ICML 2003", Proceedings of the Twenty International Conference on Machine Learning Washington, DC:[s.n.], **(2003)**, pp. 616-623.
[20] D. Wegener, M. Mock, D. Adranale, *et al.*, "Toolkit-based high-performance data mining of large data on MapReduce clusters［C］// ICDM: IEEE International Conference on Data Mining", Washington, DC: IEEE Computer Society, **(2009)**, pp. 296-301.
[21] C. Huiping, L. Lili, W. Jiandong, *etc*., "Weka Data mining platform and Reusability", vol. 44, no. 19, **(2008)**, pp. 76-79.
[22] Sogou Library, "Internet Corpus[EB/OL][2011-02-17]", www.sogou.com/labs/dl/ t.html.

 