RCES REVIEW OF COMPUTER ENGINEERING STUDIES

ISSN: 2369-0755 (Print), 2369-0763 (Online)
Vol.3, No. 1, March 2016, pp. 6-10
DOI: 10.18280/rces.030102

A publication of IIETA

IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering
http://www. iieta. org/Journals/RCES

An Improved Parallel Bayesian Text Classification Algorithm

Panpan Shen, Hao Wang, Zhouqing Meng, Zhenyu Yang, Zhaoping Zhi, Ran Jin and Aimin Yang

School of Computer Science and Information Technology, Zhejiang Wanli University, Ningbo, China.

Email: 1172340155@qq. com

ABSTRACT

Used the idea of cloud computing, according to MapReduce model to solve the traditional Bayesian classification algorithm suited to large-scale data deficiencies, greatly improved the speed of classification. The combination of the characteristics of the parallel algorithm was improved accordingly. Adding synonyms and word frequency filtering combined approach allows vector dimensionality reduction, reducing false positives. Wherein the particular keyword was then weighted to enhance the accuracy of the classification. Finally, the Hadoop cloud computing platform was experimentally proved that the traditional text classification algorithm after parallelization on Hadoop cloud computing platforms, has better speedup, and the improved algorithm can improve the classification accuracy.

Keywords: Cloud computing, Text classification, Parallel, Hadoop.

1. INTRODUCTION

With the development of the Internet, a large number of the data is growing rapidly, which the speed is in a geometric level. According to statistics, the amount of data on the Internet will double in every two years. So the concept of big data also came into being. Followed by a great amount of data which needs depth analysis, among them, the Web document is one of the most common and also the widest range of the data.

Automatic text categorization can improve the quality and efficiency of information retrieval, and it has been applied in many fields. Bayesian [1-4] classification method is a simple and effective method in many probabilistic classification algorithm, and in some areas shows a good performance. However, this method is not satisfactory for the classification of the document, and there will be problems when dealing with large-scale Web documents, such as the training set generates a slow speed, and the machine learning efficiency is low, and the length of the text exceeds a certain length, which resulting in a large number of errors.

At the same time, cloud computing can solve the problem of computing speed effectively, which is that the original algorithm is parallel to the original algorithm by Map/Reduce [5-8], and using the key/value method to analyze the records of processing data set. Only need to decompose the problem into a parallel operation of the sub problem, and design Map and Reduce two functions, can you use the distributed system to solve the problem without the need to consider the details.

In the context of the current big data, the traditional text classification has been gradually t suitable for the needs of users. Cloud computing can solve the problem of speed to a certain extent, but how to use cloud computing to make full use of its performance is still a worth studying problem.

In view of the above situation, in the second section, we introduce the traditional naive Bayes algorithm, and analyze the feasibility of its defects and parallelization; In the third section, the naive Bayes algorithm is improved; In the fourth section describes how to implement the algorithm on cloud computing platform through Map/Reduce model; The fifth section by using Hadoop[9-10] cloud computing platform carry out related experiments, which is proved that this method can effectively improve the classification speed and accuracy of large batch documents, and it has a faster speed and higher accuracy than the naive Bayes algorithm.

2. NAIVE BIAS CLASSIFICATION ALGORITHM AND ITS PARALLEIZATION

2.1 Naive Bias classification method

Bias classifier is a typical probability statistical classifier based on Bias theorem. The main idea is to calculate the conditional probability of each existing category for a document to be classified, and then the document is classified as the category with the highest conditional probability. The main steps are as follows:

(1)Construction of classifier

Group the training set according to the category, then the statistics of each category contains the number of features and the number of words appear. All the training set statistics save the classifier to prepare the work to complete.

(2)Classification of documents to be classified

The calculation of probability vector $(x_1, x_2, x_3, \cdots, x_n)$. feature words belonging to each category.

$$x_{k=}P\left(W_k \mid C_j\right)=\frac{1+\sum_{l=1}^{|D|} N\left(W_k, d_l\right)}{|V|+\sum_{s=1}^{|V|}\sum_{l=1}^{|D|} N\left(W_s, d_l\right)} \quad (1)$$

Among them, d_l $N\left(W_k, d_l\right)$ is d_l in W_k frequency, W_k represents a feature word, d_l represents a training text in the C_j class, $|V|$ is the total number of feature words, $|D|$ as the number of documents in the C_j class.

In order to facilitate the following description, the formula is simplified as

$$P\left(W_k \mid C_j\right)=\frac{1+T}{VC+M} \quad (2)$$

Among them, $T=\sum_{l=1}^{D} N\left(W_k, d_l\right)$ indicates the number of times that W_k appears in the C_j class. $VC=|V|$ said the general characteristics of words, $M=\sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N\left(W_s, d_i\right)$ represents the total number of features that appear in the C_j class.

Calculate the test text belong to each category of the conditional probability, and select the maximum value.

$$P\left(C_i \mid d\right)=\arg\max P\left(C_i\right)\prod_{j=1}^{m} P\left(w_j \mid C_i\right) \quad (3)$$

Among them, $P\left(C_i\right)$ is the prior probability of class C_j, m is the number of feature items.

Naive Bayes algorithm flow chart shown in Figure 1.

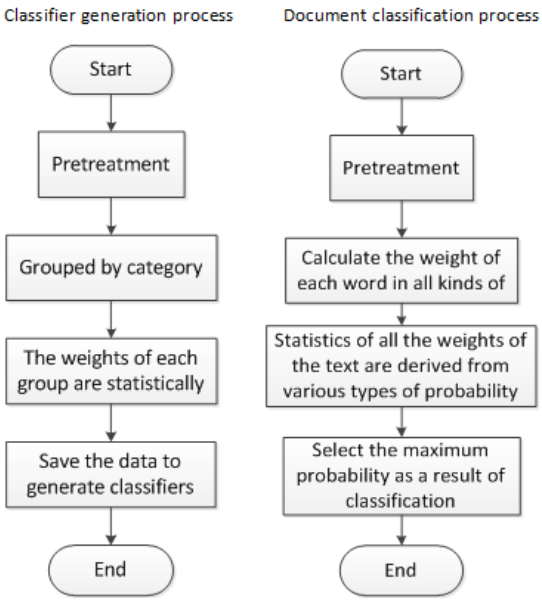

Figure 1. Naive Bayes algorithm flow chart

2.2 Naive Bayesian algorithm defe

As can be seen from the above process, the training set to generate slow speed, low efficiency of machine learning, after the text of more than a certain length will produce a large number of errors. The reasons for these problem are as follows:

(1)The accuracy of classification depends largely on the size of the training set. If the training set is too small, the classification results will be chanciness.

(2)A large number of documents will produce a large amount of features that can not be highlighted its attributes, which will fuzzy the classification results.

(3)Whether it is a great number of training set learning process, or a long document classification of statistical work,

it is required to take a considerable amount of computing resources, a single machine has been gradually unable to do the task.

2.3 The feasibility of Bias algorithm parallelization

Analyzing the process of Bayesian algorithm, we can found that whether it is the generation process of the classifier, or the process of document classification, it is made up of many independent computing. Therefore, it is the possibility of splitting and parallel computing itself. The following control figure 1 of the traditional serial Bias classification process is given in parallel after the classification of the flow chart and to be described.

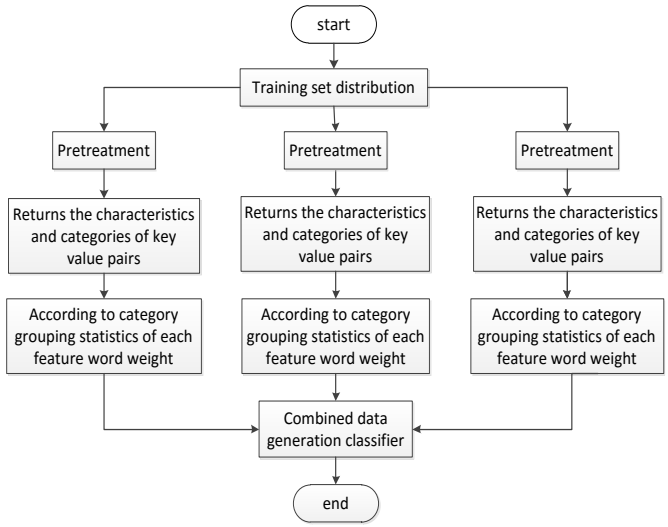

Figure 2. Classifier generated in parallel flow chart

(1) Classifier generation parallelization

The learning process of each text is the same and there is no connection. Accordingly, one of the serial learning process in parallel and the training set is divided into the training set by each node which will be trained text word segmentation statistics. The smallest unit of the slice is a text, which according to the specific number of nodes to select the appropriate size of the film. The flow chart is shown in Figure 2.

(2) Document classification process

The essence of the classification is to calculate the probability of each feature word belonging to each class and overlay, and finally get the probability of the various categories of the document and take the maximum as the classification results. The most time-consuming part of the process is the need for a large number of the $P\left(Wj \mid C_i\right)$ calculations, and the calculation process of each characteristic words is independent of each other. So the feature words one by one statistical process in parallel, each node is completed a

part of the $P(Wj \mid C_i)$ calculation. Finally, combine the output classification results. The flow chart is shown in Figure 3.

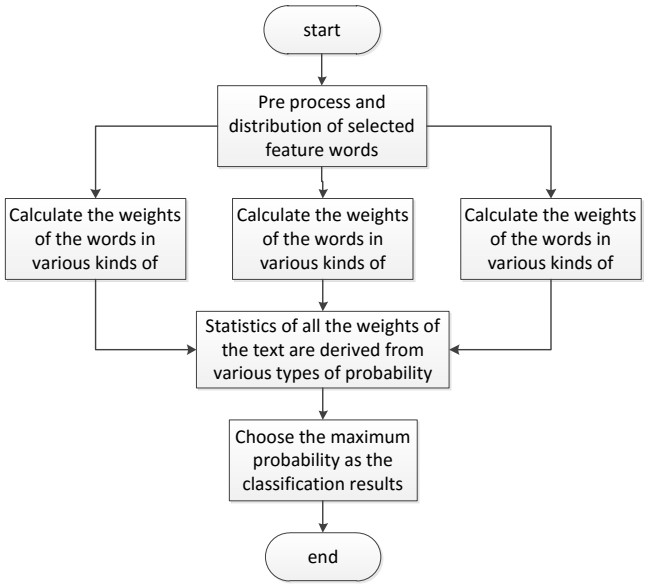

Figure 3. Parallel document classification flowchart

2.4 Improvement based on Naive Bayes algorithm

The disadvantage of the traditional Bias classification algorithm is that when the text is too large, there will be a large dimension of the vector space, which makes the classification results show discrete changes. Therefore, before the cloud computing, there are many improvements to improve the traditional Bias classification algorithm. However, the method is in serial increased all links to improve classification accuracy, although have a good effect, but for cloud computing, the classification of complex process parallel of compared with naive Bayes algorithm has improved a lot.

Therefore, this paper proposes an improved method which is suitable for the parallel improvement. The classification accuracy can be improved by the compression of a single Map/Reduce process. First filter the feature words to reduce the vector dimension, and then join the concept of the core key words to highlight the role of the core word vector.

2.4.1 Feature words selection and filtering

First, to read every word and filter out the stop words, function words, prepositions, adverbs, etc. which as the feature words recorded.

Then, the feature words are filtered and combined with the feature vector of the synonyms.

Finally, filtering out the low frequency part of feature words, the threshold setting can according to the specific training set size, the threshold of 8%. That is in every 100 training texts appear fewer than eight times the word feature which will be removed.

After a series of filtering, the vector dimension will be quite large.

2.4.2 Core key words

In the actual reading and writing, people pay more attention to the title, abstract, appeared the first and last sentences in the entry. Therefore, this article introduces the concept of the core key words, which can strengthen the role of the above entry in the classification.

In the training, separate statistics for entry in an important position, such as a title or an abstract. The results are also referred to as the filter and the filtration thresholds can appropriate to reduce. If the number of training texts is small, it can be set to 0%. After filtering is the core of the class of key words, and record the word frequency n.

When a feature word w_k belongs to the core key-words, The total frequency multiplied by $\left\lfloor \frac{n}{10} \right\rfloor \sqrt{n}$, That is $\left\lfloor \frac{n}{10} \right\rfloor \sqrt{n} \times \sum_{l=1}^{D} N(W_k, d_l)$, And then continue to calculate the posterior probability according to the naive Bayes algorithm.

Thus the weight of the key words to highlight the influence of the vector, strengthen the classification effect.

3. CLASSIFICATION ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT

3.1 Parallel generation of classifiers

Adding weighted value $k = \left\lfloor \frac{n}{10} \right\rfloor \sqrt{n}$ based on the original formula, in brief,

$$P(W_k \mid C_j) = \frac{1 + T \times k}{VC + M} \tag{4}$$

Calculation task is completed on the above 4 variables T, K, VC, M statistics, which will be completed by the two calculation.

The training text input platform, (key, value) corresponds to (text type, text content).

Map task: first of all, each category to establish the corresponding directory, and to create a corresponding sub directory, what is used to the core of the statistical key words. For all Value call word segmentation tool (here open source is very easy to transform the Chinese word segmentation plug-in, on this basis to add a filter, used to achieve feature words filtering and core keyword filtering function). After filtering, the following steps are performed each generation of a feature word:

Output intermediate results $\langle (C_j, w_i), 1 \rangle$

Output intermediate results $\langle (C_j, count), 1 \rangle$

Output intermediate results $\langle w_i, 1 \rangle$

If you read the core flag on the output of the intermediate results $\langle (C_j, w_i, core), 1 \rangle$, Core, which represents the core of key words.

Combiner function:The function is the function of the intermediate results generated by the map function in the first merger, which in order to reduce the amount of data sent to each node to the network and to reduce the burden on the network.

(1)For the set of $\langle (C_j, w_i), 1 \rangle$, if the key value is the same, then summing output the result $\langle (C_j, w_i), n \rangle$, n is representative of summation result.

(2)For the set of $\langle(C_j, count), 1\rangle$, if the key value is the same, then summing output the result $\langle(C_j, count), m\rangle$, n is representative of summation result.

(3)For the set of $\langle(C_j, w_i, core), 1\rangle$, if the key value is the same, then summing output the result $\langle(C_j, w_i, core), num\rangle$, n is representative of summation result.

(4)For $\langle w_i, 1\rangle$ set in the same key value directly deleted, only the output of the remaining pairs.

Reduce task:Statistics from each node to receive the intermediate results, and get the desired variables. Its steps are as follows:

(1)For $\langle(C_j, w_i), n\rangle$ set, if the key values are the same then summed, you can get a certain number of times wi word appears in the feature class of the document T.

(2)For $\langle(C_j, w_i, core), num\rangle$ set, if the key values are the same then summed, to obtain a core keywords Frequency n.

(3)For $\langle(C_j, count), m\rangle$ set, if the key value is the same as the sum (here count represents only a count variable, so the intermediate results generated by each feature key word is the same), that is certain to give the total number of all feature words M.

(4)For $\langle w_i, 1\rangle$ set, delete key values are the same, the key statistics on the number of remaining to give the training features in the total number of vocabulary words VC.

At this point, four of the variables T, K, VC, M statistics is completed, training set learning is completedand the classification model is established.

3.2 Parallel test text categorization

The test text for the same word filtering, the various characteristics of the word wi input classification model, (value, key) corresponding to (test text number, WI).

Map task: Input (test text number, WI)and take the appropriate parameters T, K, VC, M from the classification model. Calculate the conditional probability $P = P(W_k \mid C_j) = \frac{1 + T \times k}{VC + M}$, finally the output of intermediate results (< test text number,), P>.

Reduce task:The < (test text number, C_j), P>, Key value of the same P product. Get the posterior probability of the document for each category $P(C_i \mid d)$. And the maximum value is taken as the classification result.

4. EXPERIMENTAL RESULTS AND ANALYSIS ON CLOUD PLARFORM

The test environment for the LAN, the platform composed by Hadoop, the computer is configured to CPU i5 memory, 8G, Linux operating system. Among them, 1 is the main node server, other 9 is a child node server, and Cluster configuration is shown in Figure 4.

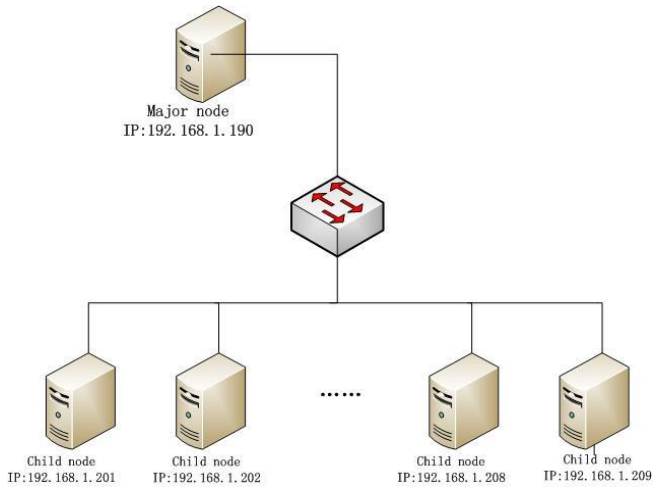

Figure 4. Cluster structure diagram

The internet corpus Sougou experimental data by Sogou laboratory provides training and testing, a total of 10 categories: automotive, financial, IT, health, sports, tourism, education, employment, culture, military.

It can be found that with the increase of the number of nodes, the processing time for the same scale data has been significantly reduced, and the improved method has a faster computing speed. It shows that the parallel computing method of the cloud computing platform can greatly improve the efficiency of the classification of large quantities of documents.

(1)Accelerated performance comparison

We used 1, 3, 5, 7, 10 nodes to test the cross classification of the data set. The classification method Pusu Bias classification method and the improved experiment is carried out, the results are shown in Figure 5.

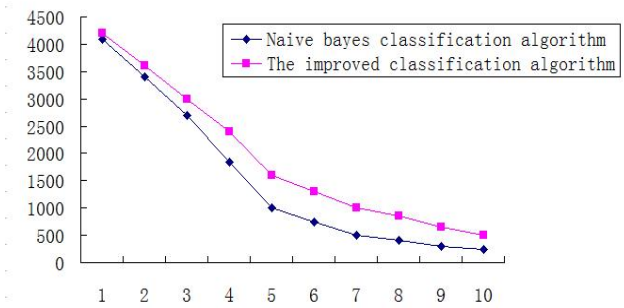

Figure 5. Calculation time contrast statistical chart

Careful analysis of the experimental data we can found that, with the increase of the number of nodes, the improved method compared with the naive Bayesian method to improve its speed is more and more obvious. This is because the method of combining feature words through the filter can effectively reduce the vector dimension and speed up the calculation of the speed. But in another way the core word statistics also slightly increase the burden of computing. So in the case of single node, the speed of the two methods is not much different.

However, the statistical calculation of the core words only accounts for a small proportion of the total amount of computation. With the increase of the number of nodes, the effect will gradually become smaller. Therefore, you can see

the difference in the speed of the two methods is gradually expanding. Until the number of nodes increases to a certain degree, the computing time of the classification becomes very short, and the gap between them is gradually reduced.

Classification effect comparison

From the existing research results [2] Pusu Bias classification recognition classification of degree 86. 1%, here no longer. The total recognition rate of the improved classification method was 91. 2%. The results of classification are listed in Table 1. It shows that the improved method can improve the accuracy of classification to a certain extent.

Table 1. Improved classification experiment results

Classification prediction	car	Sports	healthy	Tourism	IT	recruit	education	Culture	Economics	military	recall/%
automobile	7843	3	10	21	43	2	14	7	54	3	98. 0
Sports	7	7723	15	29	37	56	28	80	11	8	96. 5
healthy	27	16	7185	77	124	56	325	117	6	67	89. 8
Tourism	31	24	19	7417	66	78	68	147	114	36	92. 7
IT	5	12	77	147	7389	54	214	4	76	22	92. 4
recruit	7	5	141	22	76	6997	445	128	142	37	87. 5
education	2	15	263	7	101	24	7551	9	14	14	94. 4
Culture	44	16	347	112	76	54	976	6123	185	67	76. 5
Economics	201	3	17	137	303	38	179	44	7012	66	87. 7
military	18	5	14	28	45	22	54	93	24	7697	96. 2
recall/%	95. 8	98. 7	88. 8	92. 8	89. 5	94. 8	76. 6	90. 7	91. 8	96. 0	

5. CONCLUSIONS

In this paper, the algorithm based on the naive Bias classification algorithm based on the characteristics of distributed computing has been improved, and deployed in the Hadoop cloud computing platform, the corresponding testing and improvement. The experiments show that the improved method can effectively improve the accuracy and the speed of the classification algorithm in dealing with the mass data, which compared with the naive Bayes method.

ACKNOWLEDGMENT

This work was supported by the Ningbo Natural Science Foundation under grant No. 2015A610141, and by the Zhejiang Science and Technology Program under grant No. 2016C33195, and by the National Undergraduate Training Programs for Innovation and Entrepreneurship under grant No. 201510876021.

REFERENCES

[1] Jing Y. S., Pavlovic V., Rehg J. M., "Boosted Bayesian network classifiers," *Machine Learning*, 2008, vol. 73, no. 2, pp. 155-184.

[2] Webb G. I., Boughton J. R., Zheng F., et al. "Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification," *Machine Learning*, 2012, vol. 86, no. 2, pp. 233-272

[3] Tillman R. E., "Structure learning with independent non-identically distributed data," *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, 2009, pp. 1041-1048.

[4] Su J., Zhang H., Ling C. X., et al., "Discriminative parameter learning for Bayesian networks," *Proceeding of the 25th International Conference on Machine Learning Helsinki*, Finland, 2008, pp. 1014-1023.

[5] Ekanayake J., Li H., Zhang B., et al. "Twister: A runtime for interactive MapReduce," *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, Chicago, Illinois, USA, 2010, pp. 810-818.

[6] Dean J, Ghemawat S. Mapreduce, "Smiplified data processing on large clusters," *Proceedings of the 6th Symposium on Operating System Design and Implementation*, SanFrancisco, California, USA: USENIX Association, 2004, pp. 137-150.

[7] Thusoo A., Sarms J. S., Jain N., et al., "Hive: A warehousing solution over a map-reduce framework," *Proceedings of the Conference on Very Large Databases*, Ly-on, France, 2009, pp. 1626-1629.

[8] Dean J., Ghemawat S., "Map/Reduce advantages over parallel databases include storage-system independence and fine-grain fault tolerance for large jobs," *Communications of the ACM*, vol. 53, no. 1, pp. 72-77.

[9] Dittrich J., Quiane-Ruiz J. A., Jindal A., et al., "Hadoop++: Making a yellow elephant run like a cheetah(without it evennoticing)," *Proceedings of the VLDB Endowment*, vol. 3, no. 1, pp. 518-529, 2010.

[10] Bu Y., Howe B., Balazinska M., et al., "HaLoop: Efficient iterative data processing on large clusters," *Proceedings of the VLDB Endowment*, vol. 3, no. 1, pp. 285-296, 2010.