Parallel Processing of Improved KNN Text Classification Algorithm Based on Hadoop

Shaobo Du
School of Computer & Information Engineering
Guizhou University of Commerce
Guiyan, China
e-mail: dushaobo16@163.com

Jing Li
School of Computer & Information Engineering
Guizhou University of Commerce
Guiyan, China
e-mail: heruihao@126.com

***Abstract*—With the rapid development of mobile Internet, the network has become an important medium for people to exchange information. The research on text classification has practical significance. Using the Hadoop platform to parallelize the KNN classification algorithm can quickly and accurately classify the text, but when calculating the similarity or distance of the sample points, the KNN algorithm will increase with the increase of the sample data, which will lead to the algorithm time. Increased complexity and reduced classification accuracy. Therefore, Parallel Processing of Improved KNN text classification algorithm based on Hadoop platform is proposed. The CLARA clustering algorithm is used to cut out the samples with low similarity in the dataset, and the calculation of sample distance in the dataset is reduced. Then design the parallel KNN MapReduce program to classify the network public opinion data. The experimental results show that the improved parallel KNN algorithm improves the accuracy and time of text classification.**

Keywords-kNN; CLARA classification; public opinion; mapreduce

I. INTRODUCTION

With the rapid development of the mobile Internet, social platforms such as Weibo, blogs, and Twitter have become important media for people to obtain information, so the amount of data on the social platform is increasing. On the social platform, there are also some bad information that affects social stability. Therefore, it is of great practical significance to analyze and monitor sensitive data on social platforms in a timely manner, and to classify, warn and guide different topics. The network public opinion data has the characteristics of large quantity, unstructured and strong dispersion, and the network public opinion data is usually transmitted in the network in the form of text. The traditional text classification algorithm can better deal with the classification of text data, but when the amount of text data increases gradually, the traditional text classification algorithm can't classify the network public opinion data efficiently and quickly when dealing with large-scale network public opinion data.

Traditional classification algorithms: Naive Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). Among them, KNN classification algorithm has the advantages of simple algorithm principle, mature theory, high classification accuracy and easy implementation, so it is widely used in various fields. The KNN classification algorithm takes a long time to calculate the similarity or distance of each sample point in the data set, which leads to an increase in the time complexity of the classification algorithm. On the other hand, when the data distribution in the data set is more dispersed, the classification accuracy will be reduced. Therefore, it is a hot research topic to study how to improve the classification accuracy and time complexity of the algorithm.

At present, many scholars have conducted related research on the KNN classification algorithm. Wang Yanfei [1] proposed cutting the entire training data set based on the sample density and clustering the cropped sample data. Ma Bin et al [2] and DU Shaobo [3] parallelize the KNN classification algorithm with the Hadoop platform, and can process large quantities of data better after parallelization. Ma Ying et al [4] proposed using K-medoids clustering algorithm to crop the training set, remove the low similarity part of the sample, and then perform KNN parallelization. The experimental results show that the method can reduce the running time of the algorithm. Although the K-medoids clustering algorithm is less sensitive to outliers and noise data, but it can't process large amounts of data.

In order to solve the problem that the K-medoids clustering algorithm can't process large-volume data and improve the classification accuracy of the algorithm, the CLARA (Clustering LARGE Applications) clustering algorithm is used to tailor the parts with lower similarity or distance in the dataset; then MapReduce is used to parallelize the KNN algorithm on Hadoop platform for data classification.

II. KNN ALGORITHM PRINCIPLE

The K-Nearest Neighbor (KNN) algorithm [5], originally proposed by Cover and Hart in 1968, is very simple and intuitive, easy to implement quickly, and one of the simplest machine learning algorithms. The idea of the algorithm is that if the majority of the samples of the k most similar in the feature space (ie, the nearest neighbor in the feature space) belong to a certain category, the sample also belongs to this category.

The KNN classifier is a passive (lazy) type of learner. The pairing of the model is very simple (only need to store the training data) [6-7]. When the test data is received, the classification model is constructed, the test data is preprocessed and the components are calculated. The distance is classified according to the test data and the training data. The KNN algorithm needs to calculate the distance between the training data and the test data when

978-1-7281-0426-3/19/$31.00 ©2019 IEEE

classifying. Therefore, the algorithm consumes more resources, and each data node has independence for the similarity calculation, so the algorithm is suitable for running in a parallel environment.

Suppose the training set is L, $C_1, C_2, \cdots, C_N$ indicates that there are N categories, the total number of L training sets is M, and the feature vector dimension threshold is n. $d_i = \{x_{i1}, x_{i2}, \cdots, x_{ij}, \cdots, x_{in}\}$ $(0 < i \le M)$ represents the eigenvector form of a text in the training set L, and x_{ij} represents the weight of the j $(0 < j \le n)$ dimension of d_i. The eigenvector form of the test set is $d = \{X_1, X_2, \cdots, X_j, \cdots X_n\}$, where X_j represents the weight of the j $(0 < j \le n)$ dimension of d. Common methods used to determine the distance between a test set and each object in the training set: Cosine similarity and Euclidean distance.

The cosine similarity calculation is as shown in equation (1).

$$Sim(d, d_i) = \frac{\sum_{j=1}^{n} (X_j x_{ij})}{\sqrt{\sum_{j=1}^{n} (X_j^2)} \sqrt{\sum_{j=1}^{n} (x_{ij}^2)}} \quad (1)$$

After the K nearest neighbor texts of the test data set classification text are found by the formula (1), finally, the weight of the text to be classified d to be categorized is calculated by formula (2), and the text to be categorized is categorized into the category with the greatest weight.

$$W(d, C_j) = \sum_{i=1}^{K} Sim(d, d_i) y(d_i, C_j) \quad (2)$$

In formula (2), $y(d_i, C_j)$ is a class attribute function, as shown in formula (3).

$$y(d_i, C_j) = \begin{cases} 1, d_i \in C_j \\ 0, d_i \notin C_j \end{cases} \quad (3)$$

III. IMPROVEMENT OF KNN ALGORITHM

A flow chart for designing an improved KNN text classification algorithm based on the Hadoop platform is shown in Figure 1.

K-nearest neighbor algorithm is simple and easy to implement, but it takes a lot of time to calculate the distance of each sample point in the data set, which reduces the efficiency of the classification algorithm. So literature [4] propose K-medoids clustering algorithm to reduce the redundancy of similarity calculation by tailoring the data with low similarity or distance. K-medoids clustering algorithm does not perform well in processing large amounts of data. Therefore, CLARA clustering algorithm is introduced to solve this problem. Based on Partitioning Around Mediods (PAM) algorithm, CLARA adopts random sampling method to reduce the time complexity of computing on large data sets. Partitioning Around Medoids algorithm is a typical implementation of K-medoids clustering. The central point of cluster in PAM algorithm is a real sample point, not a central point calculated by distance. Like k-means, PAM uses greedy strategy to process clustering process.

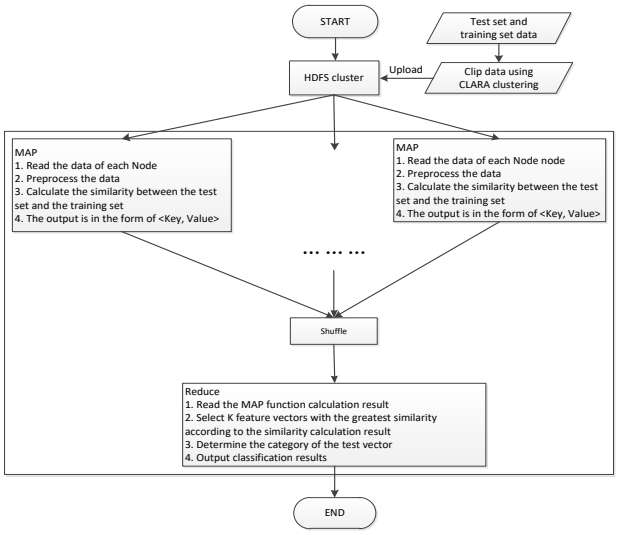

Figure 1. Algorithm framework.

Specific implementation process of CLARA algorithm[8]:

(1) Make m iterations and execute (2)-(6) steps iteratively.

(2) Extracting the same number of r objects from the whole data set D according to random sampling method to get the sample data set $S_i, S_i = (s_1, s_2, \cdots, s_r)$.

(3) Apply PAM algorithm on the sample data set S_i to find the optimal set $C_i, C_i = (c_1, c_2, \cdots, c_k)$ of k centers in the sample data set.

(4) According to C_i, find each object $O_j \in D$ in the entire data set D. at the nearest central point of Euclidean distance in C_i, and O_j is divided into corresponding clusters.

(5) Calculate the average dissimilarity of each object $O_j \in D$ in data set D according to formula $\sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i) / n$, as the evaluation standard

(6) Return to step 1 and start the next iteration.

(7) After the iteration is completed, the average dissimilarity is used as the evaluation standard to measure the clustering effect, and the smallest average disparity is the optimal clustering.

(8) Statistical sample data sets and k clustering average dissimilarity, if $Sim(D, O_i)$ is less than the given threshold,

it will be cut off from the sample data set, otherwise the sample contained in the cluster will be added to the sample set.

IV. PARALLELIZATION OF KNN ALGORITHM

Compared with existing MapReduce, KNN parallel MapReduce text categorization algorithm speeds up the execution of the algorithm, increases the pre-processing mechanism and reduces the redundancy of data sets. The implementation functions are as follows:

(1) Map function

Input: training data set and test data set, setting k value is generally odd, giving the category of training data set

Output: key-value pair <Key1, Value1>, where Key1 represents the test data set index value, and Value1 consists of the string similarity S and the category label C.

1: Method map(Key,Value,Key1,Value1)
2: {
3: for each line in Value do
the data in line is decomposed into the form of < id, x, y >.
Computational similarity $S = Sim(x, y)$;
x is the test vector and y is the training vector.
Emit(Key1,Value1);
4: }

(2) Reduce function

Input: output result of map function <Key1, Value1>

Output: <Key2, Value2>, where Key2 is the value of Key1 and Value2 is the classification result.

1: Method reduce(Key1,Value1,Key2,Value2)
2: {
3: Collection sem = new ArrayList();
4: Collection classify = new ArrayList();
5: for each v in Value1 do
construct a key-value pair $< S, C >$, where S is the similarity and C is the category label;
add the value of S to the collection sem, and C to the collection classify;
6: Sort the values in the set sem, determine K nearest neighbor sets and get the corresponding categories of the set SEM data at the same time.
7: assign the value of Key1 to Key2;
8: Emit(Key2,Value2);
9: }

The KNN algorithm is constructed into a MapReduce program to realize the parallelization of the algorithm to process the text classification. The key in the Map function is the line number of the test data set, and the Value is the training set data corresponding to the row. The data set includes the corresponding attribute fields and category labels.

The output Key1 of the Map stage represents the line number of the test data set, and Value1 represents the calculated similarity S and category label C. In the Reduce stage, Key2 represents the line number of the test data set, and Value2 represents the calculated classification result. The MapReduce programming model is shown in Figure 2.

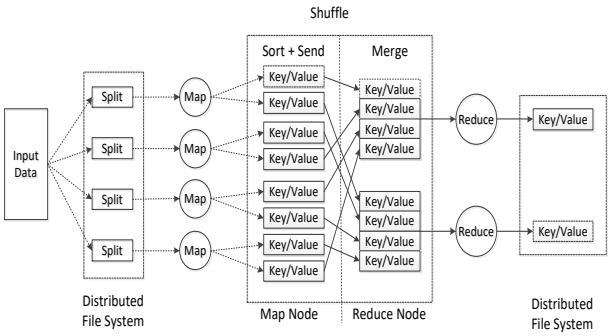

Figure 2. MapReduce framework.

V. ALGORITHM TESTING AND PERFORMANCE EVALUATION

A. Experimental Environment and Data Set Design

Firstly, the experimental data are collected through web crawler tools. The corpus is divided into two categories: positive and negative emotions. Each category contains 6,000 texts, totaling 12,000 texts. 1,000 texts were extracted from each of the two text categories, and a total of 2,000 texts were used as training sets. In order to further verify the reliability and validity of the classification algorithm, the extracted data sets are divided into test sets of different scales, as shown in TABLE I.

Use the virtual machine to build 10 Hadoop clusters, CPU: Intel E7400, 4GB RAM, CentOS 6.5, Hadoop-2.8.5. There is one Master, which realizes data upload management and scheduling and result summary of job data. The remaining nodes are Slave for distributed data storage and Computing

B. Experimental result

In order to measure the performance of the classification algorithm, the acceleration ratio, the correct rate P and the running time t of the algorithm are compared, The acceleration ratio formula is as follows:

$$Speedup(g) = \frac{Single\ node\ runtime}{Total\ node\ runtime} \quad (4)$$

Correct rate calculation formula:

$$P = \frac{Number\ of\ samples\ correctly\ classified}{Total\ sample\ size} \quad (5)$$

TABLE I. TEST SET

Test Set	Quantity	Remarks
TS1	1200	Extract 600 texts from two categories
TS2	6000	Extract 300 texts from each of the two categories
TS3	1 2000	Complete test set

Experiment 1: By comparing the accuracy and running time of KNN algorithm, literature [4] and improved KNN algorithm based on CLARA on a single node. The comparison results are shown in Table II.

TABLE II. COMPARSION OF CLASSIFICATION ALGORITHM

Test Set	Parallel KNN Classification Algorithms		literature [4]		improved KNN algorithm based on CLARA	
	P/%	*t/s*	*P/%*	*t/s*	*P/%*	*t/s*
TS1	81.7	263	80.9	211	81.3	220
TS2	83.1	405	82.8	357	84.2	365
TS3	87.7	621	89.8	558	90.5	563
Average	84.17	429.67	84.50	375.33	85.33	382.67

Table II shows that the amount of data in TS1 test set is relatively small, so the accuracy of CLARA-based KNN classification algorithm is 0.4% lower than that of KNN classification algorithm. Because CLARA clustering algorithm is an approximate algorithm for data set clipping, when the number of samples is small, it will have a certain impact on the accuracy of the algorithm. In terms of running time, the improved KNN classification algorithm based on CLARA is 10%-16% shorter than the KNN classification algorithm, while only 8.33 s more than the literature [4] classification algorithm, but the average accuracy is 0.83% higher than the literature [4] classification algorithm. When the number of samples increases gradually, the computation time of the algorithm will be shortened more and more obviously, because the redundant calculation of similarity will be reduced after the sample data is tailored.

Experiment 2: Using acceleration ratio to measure the scalability of a system. Based on CLARA improved KNN classification algorithm, this paper compares the acceleration ratio between different size data sets and different number of nodes, and the comparison results are shown in Table III.

TABLE III. ACCELERATION RATIO COMPARISON

Test Set	Number of nodes				
	1	*10*	*20*	*30*	*50*
TS1	1.0	1.86	2.87	3.92	5.01
TS2	1.0	2.00	3.00	4.02	5.13
TS3	1.0	2.05	3.09	4.11	5.15

Table III shows that the KNN text categorization algorithm based on CLARA improves linearly with the increase of the number of nodes, and the increase of the number of nodes can quickly reduce the classification time required by the categorization algorithm. This shows that the parallel algorithm based on Hadoop platform has good extensibility.

In summary, it can be seen from Experiment 1 and Experiment 2 that the parallel KNN text classification algorithm using CLARA clustering algorithm has a significant improvement in time complexity and algorithm accuracy. When the amount of data increases, the advantage of the algorithm is more intuitive improvement in the speed-up ratio, which shows that the distributed computing platform can handle a large number of data better.

VI. CONCLUSION

With the rapid development of mobile internet, the amount of network data is increasing exponentially, and the commonly used text categorization algorithms can't be processed quickly and efficiently. Therefore, parallelization of traditional algorithms has become a current research hotspot.

This paper designs an improved parallel KNN text classification algorithm based on Hadoop platform. Firstly, CLARA clustering algorithm is used to tailor the data with low similarity or distance between samples, which can reduce the time of calculating similarity or distance between samples in the later stage of text classification algorithm. Secondly, using the distributed storage feature of HDFS, a large amount of data can be stored efficiently and quickly. At the same time, MapReduce program is used to design KNN classification algorithm, and mobile computing is used to achieve fast data processing. The simulation results show that this classification algorithm has lower time complexity and better classification accuracy compared with parallel KNN classification algorithm designed by K-medoids clustering algorithm.

ACKNOWLEDGMENT

Project support by Engineering Research Center of General Colleges and Universities in Guizhou Province (KY word [2017] 022).

REFERENCES

[1] WANG Yanfei. An improved KNN Method for Reducing the Amount of Training Samples Based on Clustering and Density [D]. *Qingdao University*,2018.

[2] MA Bin. Study of an Improved K_ Nearest Neighbor Algorithm for Network Public Opinion Classification [J]. Microelectronics & Computer,2015, 32(06):62-66+72.

[3] DU Shaobo. A Parallel k-Nearset Neighbor Network Public Opinion Classification Algorithm Based on Hadoop Platfom [J]. Video Engineering,2018, 42(03):58-62.

[4] MA Ying, ZHAO Hui, CUI Yan. Parallel processing of improved KNN classification algorithm based on Hadoop platform [J]. Journal of Changchun University,2018,39(05):484-489.

[5] B. Ma.A New Kind of Parallel K_NN Network Public Opinion Classification Algorithm Based on Hadoop Platform[J], Applied Mechanics and Materials, Vols. 644-650, pp. 2018-2021, 2014.

[6] Min-Ling Zhang,Zhi-Hua Zhou. M L-KNN : A lazy learning approach to multi-label learning[J]. Pattern Recognition,2006,40(7).

[7] Chuanwen Li , Yu Gu , Fangfang Li,et al.Moving K-Nearest Neighbor Query over Obstructed Regions[C]// Advances in Web Technologies Application,2010:29-35.

[8] Jakovits P, Srirama S N. Clustering on the cloud:reduce CLARA to MapReduce[C]//Nordic Symposium on Cloud Computing & Internet Technologies.ACM, 2013:64-71.