IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 11, 2020, accepted June 2, 2020, date of publication June 16, 2020, date of current version June 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002833

Research on Product Reviews Hot Spot Discovery Algorithm Based on Mapreduce

HAO SU, QICHENG LIU, AND CHUNXIAO MU
School of Computer Science and Control Engineering, Yantai University, Yantai 264005,China

Corresponding author: Qicheng Liu (ytliuqc@163.com)

This work was supported in part by the Natural Science Foundation of Shandong Province under Grant ZR2016FM42, in part by the Primary Research and Development Plan of Shandong Province under Grant 2016GGX109004, in part by the National Natural Science Foundation of China under Grant 61702439, and in part by the 13th Five-Year Plan Key Demonstration Project for the Innovation and Development of the Marine Economy of National Oceanic Administration under Grant YHC-ZB-P201701.

ABSTRACT In recent years, with the development of e-commerce, the scale of comment data has shown an exponential growth trend. In this paper, a product review hot spot discovery algorithm based on MapReduce–PR-HD is proposed. The algorithm uses the Vector Space Model to vectorize the text data of the reviews, and utilize the TF-IDF algorithm to calculate the position weight of the feature words, then combines the Canopy algorithm and the K-Means algorithm to achieve the hot spot discovery of product reviews. At the same time, the algorithm obtain the ability to process massive data through the MapReduce framework. Experiments demonstrate that the PR-HD algorithm has high accuracy and parallel efficiency. This allows product developers to obtain more direct and effective suggestions and feedback, which allows product developers to obtain more direct and effective suggestions and feedback.

INDEX TERMS Product reviews, hot spot discovery, MapReduce, canopy algorithm, K-means algorithm.

I. INTRODUCTION

At the age of rapid development in information technology and the popularization of Internet technology, e-commerce has gradually developing and perfecting. Nowadays, the convenience of online shopping and the busy lifestyle of modern people make people's demand for e-commerce higher and higher. This status quo not only brings opportunities to the development of e-commerce platforms, but also brings about fierce competition. It is commonly understood that online reviews can reduce consumer uncertainty about product characteristics and, therefore, have the potential to increase product demand and firm profits [1]. As a result, the major e-commerce platforms should pay much more attention to the excavation of product reviews.

The problem that many companies and scholars face together is how to get the most hidden information by quickly and accurately analyzing the large-scale data [2]. In the case of opening a popular shopping site, we can find that most of the products have a large number of comments. These huge data make it difficult for producers to get product feedback in a timely manner. Besides, the information with significant commercial value is often hidden in these large-scale data [3]. The product reviews hot spot discovery algorithm can effectively filter information and solve the problem of comment analysis that is impossible only by manpower. This algorithm firstly preprocesses the comments and then carries out text clustering, and finally finds out the key points of the comments from each different category. Through this steps, the commodity producers can quickly understand the needs of users.

The purposed product review hot spot discovery algorithm based on MapReduce—PR-HD demonstrates the effective compared with others. This algorithm combines the hotspots detection algorithm with the MapReduce distributed computing framework, as well as mines the product reviews dataset through multiple computers, which finally realizes the hot spot discovery of commodity reviews. This research has certain research value in the field of product review mining.

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

II. RELATED WORK

Driven by the fifth wave of information technology revolution and global informationization, theoretical research and practical operation of e-commerce have arisen [4]. Nowadays, the mining of product reviews has become a research hotspot that has attracted much attention [5]. The mining of product reviews is the process of collecting user comments on the

Internet as mining objects, and discovering information about all aspects of products.

In recent years, the value mining algorithm of product reviews has been continuously proposed especially in sentiment analysis [6], [7]. For example, we can use the sentiment dictionary to process and represent the product reviews, and then assigns the weights of different words in the sentiment dictionary and then uses the Bayesian classification model to classify the product reviews [8]–[10]. The above work can achieve the emotional division of product reviews, but the Naive Bayes model can obtain the best experimental results when the attributes are independent of each other. The uncertainty of the actual product reviews will greatly affect the accuracy of the algorithm. Manek*et al.* [11], Al-Smadi *et al.* [12] and Basari *et al.* [13] selected different SVM model, currently widely used in Machine learning, as the basic classifier to extract reviews value. Compared with the Bayesian algorithm, these research contents have higher accuracy and avoid ''curse of dimensionality'' in the field of product review mining [14]. However, commodity reviews often involve various aspects of products. The above literature only divides comments into two aspects according to their emotions, and the mining of comments is not comprehensive enough.

In order to fully exploit the data value of comment data hiding, Babu*et al.* [15] using K-means clustering algorithm to mine commodity reviews. It can be seen that the comment mining algorithm based on clustering has greater advantages in diversity of commodity review mining. But as the size of the data increases, the accuracy of the algorithm faces new challenges. In the subsequent research, we can obtain better results by improving the initial center point selection of the K-means algorithm [16]–[18]. It can further improve the accuracy in the data analysis of shopping malls. At the same time, various clustering algorithms have also been applied to comment mining. Shao-Hua *et al.* [19] makes use of LDA on restaurant reviews to get the useful topics. Yu *et al.* [20] examines how the Latent Dirichlet Allocation (LDA) model combined with natural language processing techniques can be used to identify hot topics from free-text customer reviews. All the above research work has greatly improved the comprehensive of comment mining, but it is still limited by the size of the data in the actual environment [21]. Another problem caused by the increase in data volume is that it is difficult to determine the number of hot spots discussed in the review data set.

In the actual environment, the clustering algorithm has complexity and the comment data has high-dimensional and sparse features. This will cause the algorithm to run longer, and even cluster conflicts. Therefore, many classic clustering algorithms have been designed in parallel by Hadoop, and many optimization algorithms have been continuously proposed. For example, the FCM algorithm is designed in parallel by Liguang and Qicheng [22] using the MapReduce framework, which can more efficiently discover hot topics on microblog. Yiming *et al.* [23] proposed an improved K-means parallel algorithm has also achieved good results. Sinha and Jana [24] combines genetic algorithm with k-means algorithm and proposed a novel clustering algorithm for distributed datasets. The above work proves that the algorithm based on MapReduce can well avoid the limitation of data size, and makes the mining of hyper-scale product review data possible. At present, in order to adapt to the real environment of product review data mining, algorithmic parallel design has gradually become one of the research contents that we need to focus on.

According to the characteristics of the current reviews, a MapReduce-based product reviews hot spot discovery algorithm-PR-HD algorithm is proposed in this paper, which aims to conduct in-depth value mining in many aspects of commodity reviews in parallel. The work of this paper mainly includes the following two points. First, PR-HD algorithm can comprehensively explore the hidden value of commodity reviews, and extend the mining of commodity reviews from simple sentiment analysis to multi-faceted hotspots discovery, thus solving the problem of insufficient comment mining. It uses the Canopy algorithm to determine the number of hotspots for commodity reviews, thus solving the problem that large-scale commodity review data cannot determine the number of hotspots. Secondly, PR-HD algorithm is based on the MapReduce framework design, which shows good performance when dealing with very large-scale data, and solves the problem that serial algorithms cannot satisfy large-scale data processing. Therefore, the PR-HD algorithm is suitable for product review data that is very large and difficult to determine the discussion hotspot.

III. RELATED KNOWLEDGE

A. TEXT VECTORIZATION

VSM (Vector Space Model) is a common method of vectorizing texts. It treats the contents of all documents as a collection of words, each of which is assigned a separate index value that points to the vector dimension of the word. The dimensions of all the words in an article constitute the vector of this article. For each word in the vector, we use the TF-IDF algorithm (Term Frequency-Inverse Document Frequency) to calculate the value of its corresponding vector dimension.

Let's assume that there are a total of N documents. The number of words used in these documents is i, which we respectively record as w_1, w_2, w_3, ..., w_i. The frequencies of these words are f_1, f_2, f_3, ..., f_i. For each word w_i, the value W_i in its corresponding dimension can be obtained by Equation 1:

$$W_i = TF_i \times IDF_i = f_i \times \log \frac{N}{DF_i} \tag{1}$$

In Equation 1, TF_i is the word frequency of the word w_i, and the corresponding value is f_i. DF_i is the document frequency of the word w_i, which refers to the number of documents containing the word w_i. IDF_i is the inverse document frequency corresponding to the word w_i, and its value is represented by $\log \frac{N}{DF_i}$.

B. CANOPY ALGORITHM

The Canopy algorithm is a practical clustering algorithm because of high speed and easy implementation. The Canopy algorithm aggregates data into different clusters by presetting two thresholds T_1 and T_2. The Canopy algorithm can only roughly divide the data and the results are not accurate enough. The canopy algorithm is described as follows:

- Put the data set into the List and input the thresholds T_1 and T_2;
- Randomly pick a point as the center point, add it to the Canopy collection, and remove the point from the List collection;
- Compare the distance between the points of the Canopy set and other points. If the distance is less than T_1, divide the two points into the same cluster and delete the point from the List; If the distance is greater than T_2, the point is added to the Canopy collection and then deleted from the List collection; If the distance is greater than T_2 and less than T_1, then this point is still saved in the List;
- The algorithm ends until the List is empty.

C. K-MEANS ALGORITHM

K-means algorithm is a partition-based clustering algorithm that divides data into k clusters by inputting value of k. K-means algorithm has the advantages of simplicity and speed, but the algorithm is greatly affected by the k value, and different initial points tend to have large differences in the results. The algorithm is described as follows:

- Randomly select k values from the data set as the initial center point;
- Select any other point, calculate its distance from the initial center point and then fall into the cluster where the nearest center point is located;
- Update the center point of the cluster, calculate a new center point;
- Repeat iteration until the cluster's error satisfies the threshold and output the clustering result.

The threshold of the K-means algorithm is given by Equation 2:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - \bar{x_i}|^2 \tag{2}$$

E is the sum of the squared errors of all points. x is every point in the set, and $\bar{x_i}$ is the average distance of each point within cluster C_i;

D. COSINE DISTANCE

In order to compare the similarity between two vectors, we need to determine the distance between them. For the characteristics of text vectors, we use the cosine distance to calculate the similarity of two vectors. We represent the cosine distance between two n-dimensional vectors $d(d_1, d_2, \ldots, d_n)$ and $c(c_1, c_2, \ldots, c_n)$as sim. The value of sim is obtained by Equation 3:

$$sim = 1 - \frac{d_1c_1 + d_2c_2 + \cdots + d_nc_n}{\sqrt{d_1^2 + d_2^2 + \cdots + d_n^2}\sqrt{c_1^2 + c_2^2 + \cdots + c_n^2}} \tag{3}$$

IV. PR-HD PARALLEL ALGORITHM DESIGN

The PR-HD algorithm is mainly divided into data preprocessing, text vectorization, determining the cluster center vector and cluster analysis. Through the PR-HD algorithm, the original large and unordered comment data can be data-normalized and extracted from it. Comment hotspots from different aspects of the product. These comment hotspots provide valuable insights for producers, sellers and consumers.

A. PREPROCESSING AND TEXT VECTORIZATION

The preprocessing and text vectorization phase of the PR-HD algorithm collects product reviews in various ways, eliminates useless text and turns the comments into a collection of words. In order to convert the text into a vector format that can be calculated, the TF-IDF algorithm is used to calculate the weight of the vocabulary, so that all comments can be converted into a vector format of$< w_1 : tfidf; w_2 : tfidf >$. w_i is the word id corresponding to the word, followed by the weight of the word, which integrates all the articles into a vector format and passes it to the next stage.

B. DETERMINING THE CLUSTER CENTER VECTOR

In order to find the approximate number of clusters in the product review data, PR-HD algorithm first performs ''rough clustering'' on the data set passed in the previous stage, and uses the Canopy algorithm to determine the number of center points (k value) needed in the next stage.

The focus of this phase is to select the appropriate thresholds T_1 and T_2. We stipulate that T_1 is greater than T_2 and the selection of the threshold should be adjusted according to the actual situation to obtain more satisfactory results. When T_1 is set higher, more vectors will belong to multiple Canopy, which makes the center points close, and the clusters are not much different; When the T_1 setting is low, the number of clusters is too large, and the clustering effect is poor. When T_2 is set high, more vectors are marked as strong marks, which reduces the number of clusters; When the T_2 setting is too small, the number of clusters will increase, and the running time of the algorithm will increase. The parallelization mechanism of the Canopy phase is that each node generates a number of Canopy in the local comment data set D_i. We summarize these Canopy and finally get k clusters.

The MapReduce framework consists of two parts: Map task and Reduce task. In the Map phase of the Canopy phase, each node randomly extracts the vector v_i in the vector set D_i of the machine as a Canopy center vector, and then generates a set of central vector canopies. We calculate the distance between v_i and other vectors, use the cosine distance sim to represent the similarity between two vectors, and output the

Algorithm 1 Map Phase of Determining the Cluster Center Vector

Input: List<vector>D_i, T_1, T_2
Output: < Key of local center vector, Value of local center vector >

```
Canopy.add(v1);
while Di! = null do
    foreach vi from Di do
        if sim(Canopy.value, v1) < T1 then
            Canopy.add(vi);
        end
        if sim(Canopy.value, vi) < T2 then
            Delete vi from Di;
        end
        foreach Canopy from Canopies do
            write(centro_id, Canopy.value);
        end
    end
end
```

local center vector $< centerid, vector >$. The Map task is described in algorithm 1.

The Reduce phase is mainly responsible for summarizing the output of local Canopy center vectors by each node in the Map phase, and executing the Canopy algorithm again to obtain the global center vector. The threshold T_3, T_4 is equivalent to T_1, T_2 by default, and the output is $< key1, value1 >$. $Key1$ is the id value of the final Canopy, and $Value1$ is the global center vector. The Reduce phase is described in algorithm 2.

Algorithm 2 Reduce Phase of Determining the Cluster Center Vector

Input: List , T_3, T_4
Output: < Key of all center vector, Value of all center vector >

```
Canopy.add(v1);
while Di! = null do
    foreach vi from D do
        if sim(Canopy.value, vi) < T3 then
            Canopy.add(vi);
        end
        if sim(Canopy.value, vi) < T4 then
            Delete vi from D;
        end
        foreach Canopy from Canopies do
            write(centro_id, Canopy.value);
        end
    end
end
```

The stage of cluster number determination solves the problem that the number of topics in the commodity review discussion cannot be determined. At this stage, the number of categories of the commodity review data set is obtained and the cluster center required for the next stage is given.

C. CLUSTERING ANALYSIS

Firstly, the cluster analysis phase of the PR-HD algorithm needs to obtain the cluster center output from the previous stage, and then clusters the K-means algorithm to obtain the final cluster. Finally, we analyze the vocabulary with higher weight in each cluster to get the hot information of the comment.

The process of K-means algorithm first takes the central vector obtained in the previous stage as the k value, and then traversing the comment vector to classify the vector into clusters that closest to the distance by calculating the distance. The cosine distance sim is used to calculate the similarity between the vectors.

In the Map phase of the K-means algorithm, the main task is to read the comment vector set D_i in the local node one by one, and calculate which center point $center[i]$ (the initial center point set is $List < Canopy >$) is closest to it. We divide it into the cluster corresponding to the nearest center vector. The $key1$ value of $< key1, value1 >$ of the output result is the id of the cluster, and the value is the corresponding vector. The description of each Map task in the Map phase is described in algorithm 3.

Algorithm 3 Map Phase of Clustering Analysis

Input: List<vector>D, $center[i]$
Output: < Cluster_id, Local comment vector >

```
foreach vi from Di do
    double min_sim = ∞, dist = 0.0;
    for i = 0 to k do
        dist = sim(vi, center[i]);
        if dist<min_sim then
            min_sim = dist;
            cluster_id = i;
        end
    end
    write(cluster_id, vector);
end
```

The Reduce phase receives the output of each Map task and summarizes it, and recalculates the new center vector corresponding to the cluster with the same id as the input of the next Map phase. The output of the Reduce stage is $< key1, value1 >$. The $key1$ is the id of the cluster, and the $value1$ is the new central vector. The description of the Reduce phase is described in algorithm 4.

The output of the Reduce phase is re-introduced as an input to the Map phase, and the algorithm enters multiple iterations until a predetermined number of times is reached or the distance between the new center vector and the original center vector is less than a certain threshold.

Algorithm 4 Reduce Phase of Clustering Analysis

Input: cluster_id, List<vector>C
Output: < Cluster_id, New center comment vector >

```
double num = values.lenght;
double sim[ ], ave[ ];
foreach v_i from D do
    for i = 0 to v_i.length do
        sum[i]+ = vector.value[i];
        avg[i] = sum[i]/num;
    end
    write(cluster_id, avg);
end
```

Take out the result of the last iteration, the content is the cluster id and the vector in each cluster. We take out the word with the highest weight in each cluster, so that we can get the key information in this cluster. By analyzing the key vocabulary of all clusters, we can get the hotspots of this product's comments, thus achieving the purpose of obtaining commodity evaluation hotspots.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to test the effect of the PR-HD algorithm on commodity hotspot discovery, we evaluated the algorithm through relevant experiments in this section. First, the experimental design uses a Hadoop cluster composed of 5 computer nodes. Based on the mobile phone review data set crawled in the web, the accuracy and scalability of the algorithm are evaluated through corresponding indicators.

A. THE DESIGN OF EXPERIMENTS

1) EXPERIMENTAL ENVIRONMENT

The experiment uses a Hadoop cluster consisting of 5 nodes. The configuration of each node is identical. The configuration is as follows: CPU is Intel(R) Core(TM) i7-7700, core number is 4, frequency is 3.6 GHz; The memory size is 16GB; Ubuntu 16.04 is installed on each computer; Hadoop version is 2.2.0; The JDK version is 1.8.0.

2) EXPERIMENTAL DATA SET

The experimental data comes from the real comment data published on the Internet, and its content is the reviews under a mobile phone on the Jingdong e-commerce platform (www.jd.com) crawled by the crawler. The scale of the experimental data is about 96,000, and its content is mainly from the real feelings of all aspects of the mobile phone. Comment hotspots mainly include 9 aspects such as screen, appearance, configuration, battery life, camera, system, logistics, call quality and after-sales. The language of the reviews is Chinese, and it is not manually annotated. The specific details of the experimental data set are shown in table 1:

In order to test the feasibility of the algorithm, 60 reviews were randomly selected from 9 different aspects. A total of 540 reviews were manually marked for accuracy detection. Each review in the new reviews dataset consists of 3 parts: id, content and the category to which the review belongs.

TABLE 1. Experimental dataset details.

Dataset	Details
Data Sources	www.jd.com
Comment Object	Mobile phone
Quantity	960 thousand
Major categories	9
Data Type	Chinese
Manually Annotated	NO

B. EVALUATION CRITERION

Whether an algorithm can effectively solve a problem often requires accuracy detection. In order to detect whether the PR-HD algorithm can successfully extract the hotspots from different aspects, we use the accuracy rate as the comment indicator of the algorithm.

TABLE 2. Confusion matrix.

	Same category	Different categories
Same cluster	TP	FP
Different clusters	FN	TN

We verify the accuracy of the PR-HD algorithm by judging whether the algorithm aggregates two vectors of the same class into the same cluster. First, we need to construct the confusion matrix as shown in Table 2 according to the relationship between different vectors:

TP is the number that the same class vector pair is correctly clustered into the same cluster; *FP* is the number that the different class vector pair is incorrectlgy into the same cluster; *FN* is the number that the same class vector pair is incorrectly clustered into the different cluster; *TN* is the number that the differnt class vector pair is correctly clustered into the different cluster.

According to the four values of the above confusion matrix, the accuracy of the algorithm can be calculated. We assume that there are a total of n comments, so there are a total of C_n^2 comment vector combinations, and each element in the confusion matrix has the relationship shown in Equation 4:

$$C_n^2 = TP + FP + FN + TN \tag{4}$$

In all the same class of comment vector pairs, the ratio of the correctly clustered comment vector pairs is called positive accuracy rate (PA). It can be obtained by Equation 5:

$$PA = \frac{TP}{TP + FN} \tag{5}$$

In all the different class of comment vector pairs, the ratio of the correctly clustered comment vector pairs is called negative accuracy rate (NA). It can be obtained by Equation 6:

$$NA = \frac{TN}{TN + FP} \tag{6}$$

After combining the values of positive correctness rate and negative accuracy rate, we use the average accuracy rate (AA) as the final evaluation of the algorithm. The value of average accuracy rate can be obtained by Equation 7:

$$AA = \frac{PA + NA}{2} \tag{7}$$

The average accuracy rate (AA) represents the accurate situation after the overall vector clustering. When the value of AA is higher, the effect of clustering is better. In general, we want to increase the average accuracy rate (AA) of the algorithm as much as possible.

Speedup is usually used to measure the parallelism of an algorithm. Let us definite the time required for one processor to complete an algorithm is T_s, and the time required for p processors to complete an algorithm is T_p, then the Speedup S is obtained by Equation 8:

$$S = \frac{T_s}{T_p} \tag{8}$$

C. RESULTS ANALYSIS

When we use PR-HD algorithm for hot spot discovery on product reviews, We need to consider the threshold which is the value of the thresholds T_1 and T_2 mentioned in Section IV-B required by the algorithm. The values of T_1 and T_2 are dynamically adjusted according to the actual situation of the product data, so the threshold needs to be determined at the beginning of the experiment. Because the cosine distance is used to calculate the text similarity, the threshold is selected between 0-1 ($T_1 > T_2$). Multiple sets of tests with commonly used values are performed on the experimental data set mentioned in section V-A2, and the results are shown below.

TABLE 3. Number of hotspots under different thresholds.

T_2 \ T_1	0.4	0.5	0.6	0.7	0.8
0.3	43	32	23	16	12
0.4		22	17	13	11
0.5			13	9	7
0.6				5	4
0.7					3

The results obtained through multiple sets of experiments are shown in Table 3. Through observation, it can be seen that when the threshold value of T_1 is 0.7 and T_2 is 0.5, the setting of this parameter is more in line with the actual situation of the review data set. Therefore, set this threshold as a parameter for subsequent experiments in this article.

1) ACCURACY ANALYSIS

First, we use the data set mentioned in section V-A2 to test the accuracy of the PR-HD algorithm. Parameter configuration is performed according to the final threshold adjustment result given, We finally set T_1 to 0.7 and T_2 to 0.5, and got the experimental results more in line with the actual situation. It is more in line with the expected result of finding out 9 hotspots. Finally, the confusion matrix constructed based on the clustering results is shown in Table 4:

TABLE 4. PR-HD algorithm confusion matrix.

	Same category	Same category
Same cluster	13843	2138
Different clusters	2087	127462

We can see from the above table that 540 comments can be combined into 145530 pairs of vectors. Among them, the value of TP is 13843, the value of FP is 2138, the value of FN is 2087, and the value of TN is 127462. We substitute these values into Equations 5, 6, and 7 to get the accuracy of the algorithm. Its results are shown in Table 5.

TABLE 5. PR-HD algorithm accuracy.

	PA	NA	AA
accuracy rate	86.8%	98.3%	92.6%

We can get from the above table that the positive accuracy rate is 86.8%, and the negative accuracy rate is 98.3%. Combined with the positive accuracy rate and negative accuracy rate, we know that the value of average accuracy rate is 92.6%. It can be concluded that the PR-HD algorithm can accurately find the key information in the product review.

In order to further verify the feasibility of the PR-HD algorithm, we choose a total of three MapReduce-based algorithms from reference [22], [23] and [24] as the comparison algorithm. Reference [22] improved VSM model, and designed a parallel fuzzy c-means algorithm for hot microblogging topics discovery (HTD-PFCM). Reference [23] proposed a novel K-means algorithm (PMCSKM) for text clustering based on the selection of initial clustering centroids on density peaks. In reference [24], Ankita proposed a novel clustering algorithm for distributed datasets, using combination of genetic algorithm (GA) with Mahalanobis distance and k-means clustering algorithm.

TABLE 6. Confusion matrix of 4 algorithms.

Algorithm	TP	FP	FN	TN
PR-DH	13843	2138	2087	127462
HTD-PFCM	12595	3422	3335	126178
Ankita's method	13240	2706	2690	126894
DPMCSKM	13092	2918	2838	126682

It is different from the method of PR-HD algorithm based on Canopy to quickly find hot spots of comments. The other three documents cannot rely on themselves to determine the number of hot spots in the review data set. Reference [22] clusters the review text by given the number of review centers and membership. Reference [23] and [24] improve the quality of comment mining by optimizing the selection of comment centers during text clustering. The confusion matrix and accuracy of the above algorithm are shown in Table 6,7:

In order to compare several algorithms more intuitively, the above data is shown in Figure 1:

TABLE 7. Accuracy of 4 algorithms.

Algorithm	PA	NA	AA
PR-HD	86.8%	98.3%	92.6%
HTD-PFCM	79.1%	97.3%	88.2%
Ankita's method	83.1%	97.9%	89.9%
DPMCSKM	82.2%	97.7%	89.3%

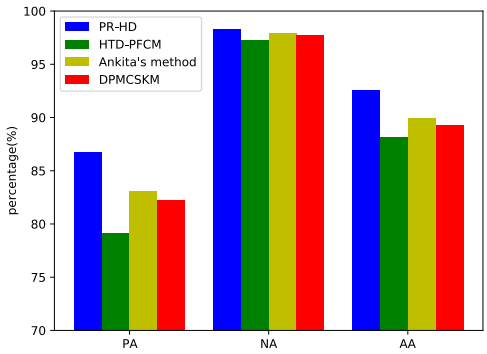

FIGURE 1. Accuracy comparison of 4 algorithm.

Table 7 and Figure 1 show the performance comparison of all algorithms. Overall, the average accuracy of the PR-HD algorithm is 92.6%, which is the highest among several algorithms. The average accuracy of HTD-PFCM algorithm is about 4.4% lower than PR-HD algorithm. This is because in the product review data set, the PR-HD algorithm uses the Canopy algorithm for pre-clustering, and selects a suitable review hot spot center from the entire review data set. In contrast, HTD-PFCM algorithm is greatly affected by the initial clustering center, and it is easy to fall into the trouble of local optimization. However, the selection of the center point in the product review data is extremely difficult. Ankita using combination of genetic algorithm (GA) with Mahalanobis distance, and considers covariance between the data points and thus provides a better representation of initial data. DPMCSKM algorithm uses the density peak to avoid the blind selection of the central point of the review data set. Therefore, the average accuracy of DPMCSKM algorithm and Ankita's algorithm is about 1.4% higher than HTD-PFCM algorithm, but they are both about 3% lower than the PR-HD algorithm.

In summary, the PR-HD algorithm is superior to other algorithms in accuracy, so it is more suitable for hot spot discovery of product review data.

2) SPEEDUP ANALYSIS

In order to measure the parallel effect of the PR-HD algorithm, we selected 3 different scales of comment data sets and completed the operations such as filter and word segmentation in advance. The resulting formatted data sizes are 362.8MB, 603.4MB, and 1.38GB. We tested the running time of these data when they are run on 1, 2, 3, 4 and 5 computers. The final result is shown in Table 8:

According to the above table, the speedup of different scale data at different nodes can be calculated by Equation 8, as shown in Table 9:

TABLE 8. Algorithm runtime(s).

Data size \ Nodes	1	2	3	4	5
362.8MB	1022.63	633.21	501.43	448.36	423.75
603.4MB	1584.36	924.31	713.62	642.43	592.94
1388.3MB	2993.16	1643.61	1224.11	1064.82	994.45

TABLE 9. Speedup of PR-HD algorithm.

Data size \ Nodes	2	3	4	5
362.8MB	1.61	2.03	2.29	2.41
603.4MB	1.71	2.22	2.47	2.67
1388.3MB	1.82	2.45	2.81	3.01

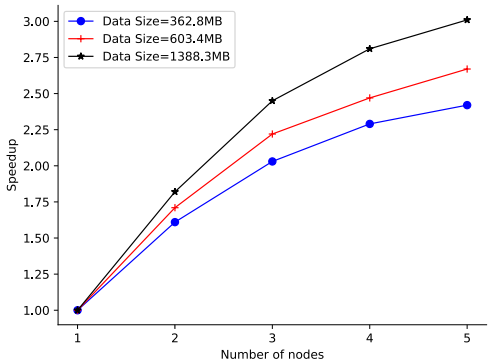

FIGURE 2. Speedup comparison of 3 data sets.

We can get the following conclusions from the table above. In case of that the data size remain unchanged, when we increase the number of nodes in the cluster, the overall performance of cluster will also increase. In case of that the data node is unchanged, when the size of the comment data set increases, the speedup also increases. The curve of the speedup is shown in Figure 2:

We can get the following conclusions from the figure above. When the number of nodes increases, the speedup of large-scale data sets will increase faster than small-scale data sets. When the data size increases, the curve of the speedup is more linear. The experimental results show that the PR-HD algorithm can effectively improve the execution efficiency of the algorithm when dealing with large-scale data. Therefore, it can be concluded that the PR-HD algorithm can meet the higher demand brought by the massive data set.

VI. CONCLUSION

A MapReduce-based product reviews hot spot discovery algorithm–PR-HD algorithm is proposed in this paper. This algorithm combines the text clustering algorithm with the MapReduce distributed computing framework, which aims to conduct in-depth value mining in many aspects of commodity reviews in parallel. We tested the PR-HD algorithm in the multi-node clusters. The experiment results show that the PR-HD algorithm has higher accuracy and can better extract the hotspots of commodity reviews to get feedback of

products in different aspects. At the same time, the PR-HD algorithm has the ability to process large-scale data, and can significantly improve the speedup when the nodes of the cluster increases, which is suitable for the mining of large-scale data.

In addition, the PR-HD algorithm avoids need to manually set the number of hotspots. But it introduces the concept of threshold. In some cases, the value of threshold will become a new factor influencing the outcome. The next step is designing a new algorithm to automatically select the appropriate threshold. It can further reduce the interference of human factors and improve the accuracy of hot spot discovery.

REFERENCES

[1] X. X. Li, L. M. Hitt, and Z. J. Zhang, "Product reviews and competition in markets for repeat purchase products," *J. Manage. Inf. Syst.*, vol. 27, no. 4, pp. 9–41, 2011.
[2] J. Dean and S. Ghemawat, "MapReduce: A flexible data processing tool," *Commun. ACM*, vol. 53, no. 1, pp. 72–77, Jan. 2010.
[3] Y. Kwark, J. Chen, and S. Raghunathan, "Online product reviews: Implications for retailers and competing manufacturers," *Inf. Syst. Res.*, vol. 25, no. 1, pp. 93–110, Mar. 2014.
[4] M. Wang, H. Zhi, and X. Li, "An empirical study of customer behavior online shopping in China," in *Proc. 7th Int. Conf. Manage. Sci. Eng. Manage.* Berlin, Germany: Springer, 2014, pp. 177–189, doi: 10.1007/978-3-642-40078-0_15.
[5] C. Dellarocas, X. Zhang, and N. F. Awad, "Exploring the value of online product reviews in forecasting sales: The case of motion pictures," *J. Interact. Marketing*, vol. 21, no. 4, pp. 23–45, Jan. 2007.
[6] X. Fang and J. Zhan, "Sentiment analysis using product review data," *J. Big Data*, vol. 2, no. 1, Dec. 2015.
[7] Z. Zhang, "Weighing stars: Aggregating online product reviews for intelligent E-commerce applications," *IEEE Intell. Syst.*, vol. 23, no. 5, pp. 42–49, Sep. 2008.
[8] C. Wei, Z. Lang, W. Huang, G. Ou, W. Yue, and D. Yang, "An empirical study of massively parallel Bayesian networks learning for sentiment extraction from unstructured text," *Commun. Pure Appl. Math.*, vol. 64, no. 7, pp. 883–919, 2011.
[9] H. Kang, S. J. Yoo, and D. Han, "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 6000–6010, Apr. 2012.
[10] J. I. Jun-Zhong, L. L. Zhang, W. U. Chen-Sheng, and W. U. Jin-Yuan, "Semantic weight-based naive Bayesian algorithm for text sentiment classification," *J. Beijing Univ. Technol.*, vol. 40, no. 12, pp. 1884–1890, 2014.
[11] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using Gini index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, Mar. 2017.
[12] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Deep recurrent neural network vs. Support vector machine for aspect-based sentiment analysis of arabic hotels' reviews," *J. Comput. Sci.*, vol. 27, pp. 386–393, Jul. 2018.
[13] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Eng.*, vol. 53, pp. 453–462, 2013.
[14] A. L. Firmino Alves, C. D. S. Baptista, A. A. Firmino, M. G. D. Oliveira, and A. C. D. Paiva, "A comparison of SVM versus naive-Bayes techniques for sentiment analysis in tweets: A case study with the 2013 FIFA confederations cup," in *Proc. 20th Brazilian Symp. Multimedia Web (WebMedia)*. New York, NY, USA: ACM, 2014, pp. 123–130.
[15] A. G. Babu, S. S. Kumari, and K. Kamakshaiah, "An experimental analysis of clustering sentiments for opinion mining," in *Proc. Int. Conf. Mach. Learn. Soft Comput. ICMLSC*. New York, NY, USA: ACM, 2017, pp. 53–57.
[16] J. Yang, Y. Ma, X. Zhang, S. Li, and Y. Zhang, "An initialization method based on hybrid distance for k-means algorithm," *Neural Comput.*, vol. 29, no. 11, pp. 3094–3117, Nov. 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/28957026
[17] K. M. Kumar and A. R. M. Reddy, "An efficient k-means clustering filtering algorithm using density based initial cluster centers," *Inf. Sci.*, vols. 418–419, pp. 286–301, Dec. 2017.
[18] G. Zhang, C. Zhang, and H. Zhang, "Improved k-means algorithm based on density canopy," *Knowl.-Based Syst.*, vol. 145, pp. 289–297, Apr. 2018.
[19] L. V. Shao-Hua, L. Yang, and H. F. Lin, "Ranks of restaurant reviews based on LDA model," *Comput. Eng.*, vol. 37, no. 19, pp. 62–64, 2011.
[20] C. Yu, X. Zhang, and H. Luo, "Mining hot topics from free-text customer reviews an LDA-based approach," in *Proc. 7th Web Inf. Syst. Appl. Conf.*, Aug. 2010, pp. 85–89.
[21] S. K. Sahu, D. P. Mahapatra, and R. C. Balabantaray, "Challenges for information retrieval in big data: Product review context," *Int. J. Comput. Appl.*, vol. 136, no. 3, pp. 27–33, Feb. 2016.
[22] F. Liguang and L. Qicheng, "Hot microblogging topics discovery based on parallel FCM," *Comput. Appl. Softw.*, vol. 32, no. 11, pp. 232–237, 2015.
[23] Y. Yiming, L. Hongzhi, and L. Haisheng, "An improved k-means text clustering algorithm based on density peaks and its parallelization," *J. Wuhan Univ. (Natural Sci. Ed.)*, vol. 65, no. 5, pp. 457–464, 2019.
[24] A. Sinha and P. K. Jana, "A hybrid MapReduce-based k-means clustering using genetic algorithm for distributed datasets," *J. Supercomput.*, vol. 74, no. 4, pp. 1562–1579, Apr. 2018.

HAO SU was born in Linyi, China, in 1995. He is currently pursuing the master's degree with Yantai University. His main research interests include cloud computing, data mining, and so on.

QICHENG LIU was born in 1970. He is currently pursuing the Ph.D. degree. He is also a Professor with Yantai University. His Research interests include cloud computing, big data, multi-agent systems, data mining, and so on.

CHUNXIAO MU was born in 1980. He is currently pursuing the master's degree. He is also a Research Assistant with Yantai University. His research interests include cloud computing, big data, and so on.