Research Article

Nan Yin*

A Big Data Analysis Method Based on Modified Collaborative Filtering Recommendation Algorithms

https://doi.org/10.1515/phys-2019-0102
Received Oct 18, 2019; accepted Nov 20, 2019

Abstract: With the rapid development of e-commerce, collaborative filtering recommendation system has been widely used in various network platforms. Using recommendation system to accurately predict customers' preferences for goods can solve the problem of information overload faced by users and improve users' dependence on the network platform. Because the recommendation system based on collaborative filtering technology has the ability to recommend more abstract or difficult to describe goods in words, the research related to collaborative filtering technology has attracted more and more attention.
According to the past research, in collaborative filtering algorithm, if Pearson correlation coefficient is used, errors will occur under special circumstances. In this study, the normal recovery similarity measure is used to modify the similarity value to correct the error value of a collaborative filtering recommendation algorithm. Based on this, a big data analysis method based on a modified collaborative filtering recommendation algorithm is proposed. This research implemented it in the cloud Hadoop environment, and measure the execution time with 2, 5 and 8 nodes. Then the research compared it with the execution time of a single machine, and analyze its speedup ratio and efficiency. The experimental results show that the execution time increases with the number of neighbors. When the number of nodes is 5 and 8, the execution time is greatly improved, which improves the efficiency of collaborative filtering algorithm and can cope with massive data in the future.

Keywords: collaborative filtering; big data; cloud environment

PACS: 89.70.Eg, 83.85.Ns, 89.75.-k

***Corresponding Author: Nan Yin:** Business School, Nanjing Xiaozhuang University, Nanjing 211171, China;
Email: yinnan123456@126.com

1 Introduction

With the progress of information technology, big data is also called large data, which refers to a large amount of information. When the amount of data is so complex that the database system cannot store, calculate, process, and analyze the information that can be interpreted in a reasonable time, it is called big data. These massive data contain useful information, such as unknown correlation, hidden patterns, potential market trends, etc., which may contain unprecedented knowledge and applications waiting to be discovered [1]. However, due to the huge amount of data and the rapid flow of data, traditional technology is often unable to conduct efficient processing and analysis, prompting relevant researchers to constantly develop a new generation of data storage equipment and technology, hoping to extract those valuable information from large data. Many companies are committed to meeting the needs of consumers. To satisfy the needs of consumers, the researcher must first understand what users need. How to recommend what consumers need or like is the most important step to satisfy the needs of consumers. The researcher can make recommendations through the habits and preferences of consumers. Quantitative data can be used as the basis for our analysis, and big data analysis has become a link closely related to life.

Due to the explosive growth of digital information and the increasing number of visitors using the network, information overload has become a potential challenge nowadays. People want to get interesting information on the network in real-time, which is also the main reason for the increasing demand for recommendation systems. The recommendation system can filter out important and useful information according to users' preferences and interests. Therefore, the recommendation system can solve the problem of information overload. In addition, the recommendation system can also predict products that may be of interest to a particular user, depending on other users who have similar preferences with that user. That is to say, the content-based filtering and collaborative filtering are com-

mon methods in the recommendation system. For users, recommendation system can greatly shorten their time to browse a large amount of information and quickly select products suitable for them; For service providers, importing recommendation system can help their customers find products of interest in real-time, so that more consumers will be willing to buy products on the service platform and become loyal customers.

With the rapid development of the Internet, it also represents that there are many open resources on the network, and the high proportion of new information increases, and there is no way to compare and analyze the filtering information, which makes it difficult for users to distinguish and filter the appropriate information, which also shows another common discussion topic of the Internet information overload. People search the resources on the network by the help of search engine, and recommendation system is a kind of concept that provides the information needed by users actively [2].

In order to meet the needs of different users in big data, recommendation algorithms are generated, among which the collaborative filtering recommendation algorithm is one of them. Current collaborative recommendation algorithms focus on the design of personal computers. In order to cope with the trend of massive data, the system can know the user's interests at the moment and meet the user's needs in time. The speed of data processing is the decisive key. The execution speed of the PC cannot meet the real-time requirement, so the combination of cloud and collaborative filtering algorithm has the value of implementation.

According to past research, in a collaborative filtering algorithm, if the Pearson correlation coefficient is used, errors will occur in special cases. In this study, the Normal Recovery Similarity Measure is used to modify the similarity value to correct the error value of the collaborative filtering recommendation algorithm, which is the basis of the collaborative filtering algorithm.

There are two main purposes of this study. The first purpose of the research is to measure the running time with 2, 5 and 8 nodes in the cloud Hadoop environment, compare with the running time of a single computer, and then analyze its acceleration and efficiency. The second purpose of the research is to analyze the prediction results by using three algorithms: the Jaccard similarity coefficient, Pearson similarity and Normal recovery similarity measure.

2 Discussions on Related Literature

2.1 Recommendation System

The recommendation system is a reference for recommending and providing consumers to buy goods. These suggestions are based on many decisions, such as what products do consumers buy? Which movie did the consumer see? Alternatively, what articles do consumers read online? Due to the explosive growth of digital information and the increasing number of visitors using the network, information overload has become a potential challenge nowadays. People want to get interesting information on the network in real-time, which is also the main reason for the increasing demand for recommendation systems. The recommendation system can filter out important and useful information according to users' preferences and interests. Therefore, the recommendation system can solve the problem of information overload. In addition, the recommendation system can also predict products that may be of interest to a particular user, depending on other users who have similar preferences with that user [3].

Recommendation system can greatly shorten the time for users to browse a large amount of information and quickly select products suitable for them. For service providers, importing recommendation system can help their customers find products of interest in real-time so that more consumers will be willing to buy products on the service platform and become loyal customers. The operation process of recommendation system is as follows: first collect user's information, including preferences and purchased products, etc., then the system will learn and build models independently, and finally predict products that users may be interested in and recommend them, while the system will collect user's selected data and go back to the first stage for repeated execution [4].

In order to reduce the additional cost of searching information, the recommendation system can recommend potential information, services or products that users may need according to their preferences, interests, behaviors or needs [5]. Recommendation system is a system that helps users filter information. Its core task is not only to filter information effectively, but also to find out users' preferences and give users interested information [6]. With the support of recommender system, the flooding of information and the complexity of online search can be reduced [7], and the convenience of searching and filtering network data can be improved.

According to different methods, common recommendation systems are divided into three types: collaborative

filtering, content filtering and knowledge-based recommendation. Content-based filtering represents the user's preferences by the characteristics of the project, summarizes the user's preferences through the user's click through records or viewing times, and finds the items that meet the preferences as recommendations [8]. The characteristic of collaborative filtering is to collect users' evaluation of the project to evaluate users' preference model, and to evaluate the possible score of the project by the same user group. The final knowledge-based recommender can explain the relationship between needs and recommended textbooks, and recommend specific textbooks to suitable users. In the process, learners contribute their own preference model, so that the recommendation system can interact with it [9].

2.2 Collaborative filtering algorithm for the recommendation system

Collaborative Filtering refers to other users' past preferences to other users based on their similar interests. The similarity between the two is calculated by each user's past score on the item, which is used to calculate the similarity between users. Collaborative filtering can be divided into user-based filtering and item-based filtering. Collaborative filtering aims at identifying other users who have similar preferences with target users, while Schafer *et al.* argues that the recommendation of people-to-people correlation refers to the relevance of users' purchases on e-commerce websites [10].

O'Donovan & Smyth [11] pointed out that collaborative filtering recommendation, also known as social filtering recommendation, is mainly based on user experience or suggestions with similar attributes or interests as the basis of providing personalized information. By recording and comparing user product or service preference data, users are divided into several communities with high degree of internal user relevance Cooperation recommendation reference. Herlocker, konstan, & Riedl [12] also mentioned that collaborative filtering system is to predict the user's preference for a certain transaction or information by connecting a group of people who have common interests with the user. Herlocker, *et al.* [13] pointed out that the operation principle of collaborative filtering is to automate the process of word-of-mouth effect, and the suggestions made by the system are based on the preferences of other users with similar preferences.

Assuming that there is a user set of N users u_i, $1 \le i \le$ N, and an item set of M items p_j, $1 \le j \le$ M, a user u_i will express his/her idea of an item p_{jj} as a score, but r_{ij} as a positive integer. Usually, the higher the score is, the more positive the user likes to give feedback. If a user u_i fails to score an item p_i, then $r_{ij} = 0$, the information is stored and expressed in the form of R:

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1M} \\ r_{21} & r_{22} & \cdots & r_{2M} \\ r_{31} & r_{32} & \cdots & r_{3M} \\ \vdots & \vdots & & \vdots \\ r_{N1} & r_{N2} & \cdots & r_{NM} \end{pmatrix} \quad (1)$$

The main purpose of collaborative filtering is to generate a list of product recommendation sequences for each user based on the information of a user's item score matrix. For this purpose, each collaborative filtering recommendation system will have an algorithm to predict the score of each user to each item. Rating is used to generate a list of recommendations.

Traditional collaborative filtering recommendation will find similar items or users according to the similarity comparison between users or objects. The most basic way is to add up and average the scores of similar users on items, and then get the scores of these users on the items, although it is reasonable and very theoretical. Effective methods, but in the actual recommendation system data, the serious sparse data makes the similarity almost impossible to complete the comparison, and a large number of users and items lead to a very time-consuming computing process.

2.3 User-based Collaborative Filtering Algorithms

User-based collaborative filtering algorithm is suitable when the number of items is much larger than that of users, and users change less; Project-based Collaborative filtering algorithm is suitable when the number of users is much larger than that of items, and the number of items changes less. Because the number of items in this experiment is large and fixed, the user-based collaborative filtering algorithm is adopted in this paper. User-based Collaborative Filtering, first proposed by Schafer *et al.* [14] refers to a recommendation based on the similarity of preferences between users. For example, recommend products that a consumer might like based on the relevance of goods purchased by other consumers on e-commerce websites. The algorithm uses all User and Item databases to predict User's Item score. The most commonly used technique is the Nearest Neighbor Method, which identifies the users who scored similar items and all scored similar items, *i.e.*

the users' neighbors. Then the user predicts these items through other items scored by neighbors and uses the Top-N recommendation method to recommend the first N items of interest.

The basic idea of user-based collaborative filtering algorithm is that if user A likes item a, user B likes item a, b, c, and user C likes item a and c, then user A is similar to user B and C because they both like a, and user who likes a likes c, so recommend C to user A. The algorithm uses the nearest-neighbor algorithm to find a user's neighbor set. The users of the set have similar preferences with the user. The algorithm predicts the user according to the neighbor's preferences.

The mathematical model of collaborative filtering recommendation algorithm can be expressed as follows: for each user, its optimization goal is:

$$J^{(j)} = \min_{\theta^{(j)}} \frac{1}{2m^{(j)}} \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} \left(\theta_k{}^{(j)}\right)^2 \quad (2)$$

Among them, the θ^j denotes the preference characteristics of the user j, x^i denotes the characteristics of the movie i, $y^{(i,j)}$ denotes the rating of the user j on the movie i, $i : r(i,j) = 1$ denotes that the user j has rating on the movie i (not missing value), and m^j denotes the number of the user j rating the movie. Since the left and right terms have m^j, the above formula can also be written as follows:

$$J^{(j)} = \min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(\theta_k{}^{(j)}\right)^2 \quad (3)$$

Then, the gradient descent is used to update θ^j, and θ^j is the preference feature of the user j.

$$J^{(j)} = \min_{\theta^{(j)}} \frac{1}{2} \sum_{i:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(\theta_k{}^{(j)}\right)^2 \quad (4)$$

If the user's preference for θ is known, then the step can learn the movie's feature x. For each movie, the optimization function is

$$J^{(i)} = \min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k{}^{(i)}\right)^2 \quad (5)$$

Then, the gradient descent is used to update x^i

$$x^{(i)} = x^{(i)} - \alpha \left[\sum_{j:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)} \right) \left(\theta^{(j)}\right)^T + \lambda \sum_{k=1}^{n} \left(x_k{}^{(i)}\right) \right] \quad (6)$$

The resulting x^i is the feature of the movie i.

2.4 Collaborative Filtering Program

The first step is similarity calculation: similarity calculation between users or projects is the key step of collaborative filtering. In collaborative filtering, common methods include cosine similarity, advanced cosine similarity and Pearson correlation coefficient.

The second step is neighbor selection: as long as different users join the neighborhood, the accuracy of prediction will change. Therefore, the researcher should carefully select some neighbor active user methods, the traditional Top-N algorithm in N-neighbor prediction. In addition, people in different countries or regions are more likely to have different preferences. Therefore, when selecting neighbors for active users, it is necessary to consider the location of users. Because of the development of mobile network, location information can be obtained by mobile client or IP address and sent to server for further analysis. Usually, users can be divided into multiple partitions according to their location. Users in the same partition have priority in neighbor selection.

The third step is prediction: based on neighborhood similarity and score, rank the scores.

The forth step is project ranking: Once the forecast is obtained, the recommendation system needs to rank all items according to the forecast score. In order to improve the diversity of suggestions, projects with larger predictions and lower popularity should rank higher.

The fifth step is selecting the first n items: After sorting all the options, the first n items are provided to the user, where n is the default parameter required before recommending the task.

2.5 Computation of Similarity

As for the calculation of similarity, the existing basic methods are based on vectors. In fact, the distance between two vectors is calculated. The closer the distance is, the greater the similarity is. In the two-dimensional user-item preference matrix of the recommended scenario, the researcher

can use a user's preference for all items as a vector to calculate the similarity between users, or use all users' preference for one item as a vector to calculate the similarity between items.

2.5.1 Pearson correlation coefficient

Pearson correlation coefficient has two concepts, one is size or strength. In terms of absolute value, the greater the absolute value, the higher the correlation between the two; the smaller the value, the lower the correlation between the two. One is the direction symbol, that is, when the coefficients are positive or negative, the relationship between the two directions changes in the positive direction, one becomes larger, one becomes smaller, and the other becomes smaller, which is called positive correlation; Negative values change in reverse, one becomes larger and the other smaller. The smaller one is, the larger the other is, which is called negative correlation. If it is zero, one becomes smaller and the other may become larger or smaller or unchanged, that is zero correlation.

Pearson correlation coefficient is generally used to calculate the degree of tightness between two fixed-distance variables, and its value is between [−1, +1]. s_x, s_y are standard deviations of x and y samples.

$$P(x,y) = \frac{\sum x_i y_i - n\overline{xy}}{(n-1)s_x s_y} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum {x_i}^2 - (\sum x_i)^2}\sqrt{n\sum {y_i}^2 - (\sum y_i)^2}} \tag{7}$$

2.5.2 Jaccard similarity coefficient

The Jaccard similarity coefficient is a statisticused for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{8}$$

If A and B are both empty, we define $J(A, B) = 1$.

$$0 \le J(A,B) \le 1 \tag{9}$$

The MinHash min-wise independent permutations locality sensitive hashing scheme may be used to efficiently compute an accurate estimate of the Jaccard similarity coefficient of pairs of sets, where each set is represented by a constant-sized signature derived from the minimum values of a hash function [15].

The Jaccard distance, which measures dissimilarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union:

$$d_J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{10}$$

3 Improvement of Collaborative Filtering Algorithms by Normal Restoration Similarity Measure

There are many different similarity algorithms in collaborative filtering algorithm. The core concept of Jacquard similarity coefficient can be seen from the following formulas:

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{11}$$

The number of items scored by user A and user B divided by the number of items scored by user A or user B falls between 0 and 1.

Pearson correlation coefficient is the most famous similarity algorithm, and its value falls between 1 and - 1. If user-based collaborative filtering is used, the formula is as follows:

$$Sim(u,v) = \frac{\sum\limits_{i \in I} \left(r_{u,i} - \overline{r}_u\right)\left(r_{v,i} - \overline{r}_u\right)}{\sqrt{\sum\limits_{i \in I} \left(r_{u,i} - \overline{r}_u\right)^2}\sqrt{\sum\limits_{i \in I} \left(r_{v,i} - \overline{r}_v\right)^2}} \tag{12}$$

I is an item with a score between user u and v. r_u and i represent user u's score for item i, r_v and i represent user v's score for item i, and $\overline{r}_u$ and $\overline{r}_v$ represent the average value of all user u's scores and the average value of all user v's scores. If the collaborative filtering is based on goods, the formula is as follows:

$$Sim(i,j) = \frac{\sum\limits_{u \in U} \left(r_{u,i} - \overline{r}_l\right)\left(r_{u,j} - r_j\right)}{\sqrt{\sum\limits_{u \in U} \left(r_{u,i} - \overline{r}_l\right)^2}} \sqrt{\sum_{u \in U} \left(r_{u,j} - r_j\right)^2} \tag{13}$$

U is an item with the same user rating between item i and j. r_u and i represent user u's rating of item i. r_u and j represent user u's rating of item j. $\overline{r}_i$ and $\overline{r}_j$ represent the average value of all item i's rating and the average value of all item j's rating.

These two collaborative filtering algorithms use the same prediction formula, and user-based collaborative filtering formula is:

$$Score_{u,i} = \frac{\sum Rating_{v,i} \cdot sim(u,v)}{\sum sim(u,v)} \tag{14}$$

The meaning of the formula is represented by: user v has a score, and user u has not scored all items multiplied by user u, v similarity, divided by the sum of user u, v similarity. The Item-based collaborative filtering formula is:

$$Score_{u,i} = \frac{\sum Rating_{u,j} \cdot sim(i,j)}{\sum sim(i,j)} \tag{15}$$

However, in some cases, errors may occur in the calculation of Pearson correlation coefficient. The following results can be obtained when calculating the similarities between user u_1 and user u_2, user u_2 and user u_3:

$$Sim(u_1, u_2) > Sim(u_3, u_2) \tag{16}$$

But in fact, the similarity between user u_2 and u_3 should be relatively high, because user u_1 scores range from 1 to 5, while user u_2 and u_3 scores range from 2 to 4. This study proposes an improved approach: using normal recovery similarity measure.

$$\begin{aligned} Sim(u,v) &= 1 - \frac{dist(\overline{u}, \overline{v})}{dist_{\max}} \\ &= 1 - \frac{\sqrt{\sum_{i \in I} (nr_{u,i} - nr_{v,i})^2}}{\sqrt{\sum_{k=1}^{|I|} (1-0)^2}} \\ &= 1 - \frac{\sqrt{\sum_{i \in I} \left(\frac{r_{u,i} - r_{u\min}}{r_{u\max} - r_{u\min}} - \frac{r_{v,i} - r_{v\min}}{r_{v\max} - r_{v\min}} \right)^2}}{\sqrt{\sum_{k=1}^{|I|} 1}} \end{aligned} \tag{17}$$

The formula is simplified as follows:

$$Sim(u,v) = 1 - \frac{\sqrt{\sum_{i \in I} \left(\frac{r_{u,i} - r_{u\min}}{r_{u\max} - r_{u\min}} - \frac{r_{v,i} - r_{v\min}}{r_{v\max} - r_{v\min}} \right)^2}}{\sqrt{|I|}} \tag{18}$$

The similarity between user u_1 and u_2 is less than that between user u_2 and u_3, and the similarity between user u_5 and u_6 is 0. The formula of the prediction score is the normal recovery similarity prediction formula:

$$\hat{r}_{u,l} = r_{u\min} + (r_{u\max} - r_{u\min}) \frac{\sum_{u' \in u} sim(u,u') \cdot nr_{u',i}}{\sum_{u' \in u} sim(u,u')} \tag{19}$$

$r_{u\min}$ is the lowest score evaluated by user u, $r_{u\max}$ is the highest score evaluated by user u, $\mathrm{Sim}(u, u')$ is the similarity between user u and user u'. In this paper, the similarity measure of normal recovery is used as the basis of collaborative filtering algorithm.

4 Experimental environment and methods

The program language used in this study is *R* data analysis language. One server and four hosts were selected as hardware cloud environment to test on 2, 5 and 8 nodes respectively. The data used in this study are from the IMDB Film Scoring Website (http://www.imdb.com). A total of 224836 score records were used [16, 17]. There are less than 20 users who delete scoring items from the data of this experiment, and all users have the same score. Because the accuracy of collaborative filtering algorithm will increase with the increase of the value of k, the neighborhood k is tested from 1 to 10 in the experiment process [18].

As a user-based collaborative filtering algorithm, the experimental structure is divided into four parts: (1) calculating the maximum and minimum scores of all users; (2) calculating the similarity of all users; (3) calculating the prediction scores. (4) In another experiment, the same data was used to recommend the item with the highest prediction score, and the number of neighbors used was 3. In this study, three different algorithms are used for prediction, namely, the Jaccard similarity coefficient, Pearson similarity and Normal recovery similarity measure.

5 Research results and analysis

The experiment first calculates the execution time of a single personal computer. As can be seen from Figure 1, where the abscissa k is the number of neighbors, when the value of k increases, the running time will be greatly increased, because according to the formula, when the value of k increases, the time will be exponential growth.

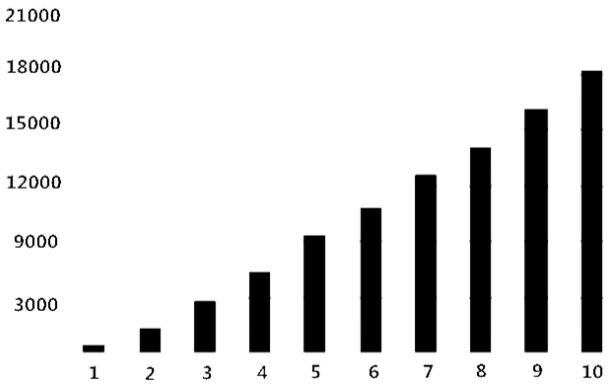

Figure 1: The running time of a personal computer (the number of neighbors in abscissa K)

Table 1: Efficiency comparison of personal computers with 2, 5 and 8 nodes

K	PC	2 nodes	Acceleration Ratio	5 nodes	Acceleration Ratio	8 nodes	Acceleration Ratio
1	721	1146	0.624	529	1.452	345	2.445
2	2245	3046	0.654	1391	1.539	879	2.489
3	4456	6234	0.691	3156	1.482	1846	2.546
4	7397	11862	0.663	5256	1.383	2875	2.583
5	10695	15672	0.647	7145	1.584	4489	2.498
6	12341	22478	0.586	8763	1.389	5446	2.437
7	14619	22478	0.642	12189	1.498	6450	2.510
8	12147	32458	0.545	12487	1.587	7215	2.674
9	22462	36542	0.629	15655	1.445	8889	2.348
10	25246	35425	0.542	14586	1.478	11241	2.457

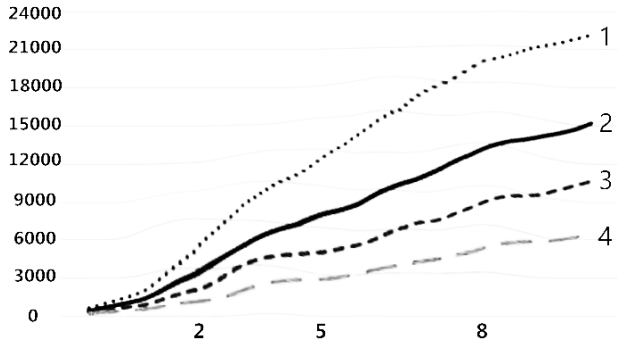

Figure 2: Comparisons of running time between PC and Hadoop with 2, 5 and 8 nodes

Because Hadoop's hardware environment consists of three hosts, it corresponds to two, five and eight nodes [19]. Table 1 compares the performance of the PC with that of the two nodes in the case of adjusting the k value (k = 1-10). At two nodes, it happens to be executed by one host. Compared with the execution of personal computer, it has more time to transmit and configure, so the execution efficiency is not good.

In Table 1, the performance of the PC with 5 nodes and 8 nodes is significantly improved compared with that of the PC with 5 nodes and 8 nodes when the K value is adjusted. In the case of five nodes, it can be seen that the acceleration ratio is greater than 1, which means that the execution speed of five nodes is about 0.5 times faster than that of a single computer [20]. In the case of 8 nodes, it can be seen that the acceleration ratio has been increased to more than 2 times, about 2.5 times, and the maximum acceleration ratio is 2.67 times when the number of neighbors k equals 4.

Figure 2 shows a comparison of running time curves of 2, 5 and 8 computing nodes between PC and Hadoop. From Figure 2, it can be seen that when the number of hardware resources and nodes in cloud environment is too low (Curve 1), it is not suitable for cloud execution. However, when the number of hardware resources and nodes in cloud environment is increased (Curve 3, 4), the collaborative filtering algorithm can effectively accelerate the calculation.

The formula used for calculating the acceleration ratio is speedup=T_a/T_b,

T_a represents the running time of a personal computer, T_b represents Hadoop runtime.

In another experiment, three different algorithms are used to calculate the result prediction. The experimental results are completely consistent. It can be speculated that there are two reasons for this result. The first one is the data set. Because the data source used in this experiment is the score of the website, it depends on the rater's interests, so the matrix is sparse in numbers [21]. Users may only want to evaluate their favorite projects, resulting in positive correlation of similarity, so the calculation of similarity will have similar results. The second reason is that this research only recommend the highest project, so other possible projects may be ignored.

Table 2 is part of the recommendation results of three different algorithms. The results are expressed by the first 10 users out of 100 users. The contents of the table are movie numbers. Each column represents different users. From the table, it can be seen that the recommendation results of each user in three different algorithms are the same.

Table 2: Top 10 Recommended Results of the Three Algorithms

Users	Jaccard similarity coefficient	Normal Restoration Similarity Measure	Pearson similarity
1	925468	925468	925468
2	24589	24589	24589
3	252465	252465	252465
4	52245774	52245774	52245774
5	52547	52547	52547
6	38625	38625	38625
7	3545562	3545562	3545562
8	2542588	2542588	2542588
9	75225	75225	75225
10	855265	855265	855265

6 Research conclusions

With the increasingly frequent e-commerce transactions nowadays, more and more sellers choose to sell goods online, which also brings a huge number of goods. In the past, the collaborative filtering recommendation system will treat each item as a feature to calculate, but in today's data form, it is unrealistic and massive. Users and commodities also bring about the problem of extremely sparse data, resulting in the recommendation system operation speed is too slow, or even unable to work.

With the advent of cloud era, data growth rate is very fast. In a massive data environment, when the researcher need to find solutions to problems, execution speed will be the key. In this paper, a collaborative filtering algorithm modified by normal recovery similarity measure is adopted, and the speed is improved by 2.67 times through the cloud environment simulation. With the increase of actual data, the operation of personal computers will take more time, and the ability to store data will be limited to a certain extent. Using MapReduce on Hadoop distributed platform to distribute operation and data to different hosts can save a lot of time and data burden. Hadoop's Distributed File System (HDFS) guarantees the correctness of the data and restores the similarity measure normally. After modification, its prediction accuracy is improved. The experimental results show that the execution time increases with the number of neighbors. When the number of nodes is 5 and 8, the execution time is greatly improved, which improves the efficiency of collaborative filtering algorithm and can cope with massive data in the future.

However, there are some shortcomings in this study. For example, when the collaborative filtering algorithm is faced with sparse matrix distribution, it will make prediction difficult. In the follow-up study, the researcher can try to find other recommended algorithms and improvement directions, such as the construction of the multi-agent model combined with neural network and collaborative filtering algorithm. Nowadays, with the increasing amount of data, using R language to analyze data in massive data will encounter layer-by-layer obstacles, too long analysis time, insufficient memory and so on. Using the methods of Hadoop Distributed File System (HDFS) and Map Reduce in Apache Hadoop Open Source Software can improve computing efficiency and storage space management and increase capacity.

References

[1] D'Angeac G.D., Big data: the management revolution, Harvard Business Review, 2012, 90(10), 60-68.
[2] Xu H.L., Wu X., Li X.D., Yan B.P., Comparison study of internet recommendation system: comparison study of internet recommendation system, Journal of Software, 2009, 20(2), 350-362.
[3] Feng L., Guo W., Yu D., Gao Q., Gao K., Xue, Z., et al., Classification of different therapeutic responses of major depressive disorder with multivariate pattern analysis method based on structural mr scans, Plos One, 2012, 7(7), 1-11.
[4] Karabadji N.E.I., Beldjoudi S., Seridi H., Aridhi S., Dhifli W. Improving memory-based user collaborative filtering with evolutionary multi-objective optimization, Expert System with Applications, 2018, 98, 153-165.
[5] Rashid A.M., Albert I., Cosley D., Lam S.K., McNee S.M., Konstan J.A., Riedl J., Getting to know you: learning new user preferences in recommender systems, Proceedings of the 7th International Conference on Intelligent User Interfaces (January 13 - 16, 2002, San Francisco, CA, USA), ACM New York, 2002, 127-134.
[6] Liang T.P., Lai H.J., Ku Y.C., Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings, Journal of Management Information Systems, 2006, 23(3), 45-70.
[7] Xiao B., Benbasat I., E-commerce product recommendation agents: Use, characteristics, and impact, Mis Quarterly, 2007, 31(1), 137-209.
[8] De Meo P., Quattrone G., Terracina G., Ursino D., An XML-based multiagent system for supporting online recruitment services, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on Systems Man & Cybernetics Part A Systems & Humans, 2007, 37(4), 464-480.
[9] Yoshii K., Goto M., Komatani K., Ogata T., Okuno H.G. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model, IEEE T Audio Speech, 2008, 16(2), 435-447.
[10] Li J., Kai Z., Yang X., Peng W., Jie W., Mitra K., et al., Category preferred canopy-k-means based collaborative filtering algorithm, Future Generation Computer Systems, 2018, 93, 1046-1054.
[11] O'Donovan J., Smyth B., Trust in recommender systems, Proceedings of the 10th international conference on Intelligent user

interfaces (January 09 - 12, 2005, San Diego, CA, USA), ACM New York, 2005, 167-174.
[12] Herlocker J.L., Konstan J.A., Riedl J., Explaining collaborative filtering recommendations, Proceedings of the 2000 ACM Conference on Computer Supported Cooperative work (December 02 - 06, 2000, Philadelphia, PA, USA), ACM New York, 2000, 241-250.
[13] Herlocker J.L., Konstan J.A., Borchers A., Riedl J., An algorithmic framework for performing collaborative filtering, Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (August 15-19, 1999, Berkeley, CA, USA), ACM New York, 1999, 230-237.
[14] Good N., Schafer J.B., Konstan J.A., Borchers A., Sarwar B., Herlocker J. et al., Combining collaborative filtering with personal agents for better recommendations, 1999, 439-446.
[15] Edith C., Min-Hash Sketches, Springer New York, 2016.
[16] Cantero A., Crespo F., Ferrer S., The triaxiality role in the spin-orbit dynamics of a rigid body, Applied Mathematics & Nonlinear Sciences, 2018, 3, 187-208.
[17] Gao W., Wang W., A tight neighborhood union condition on fractional-critical deleted graphs, Colloquium Mathematicum, 2017, 149, 291-298.
[18] Gao W., Wang W., New isolated toughness condition for fractional-critical graph, Colloquium Mathematicum, 2017, 147, 55-65.
[19] Khalique C.M., Mhlanga I.E., Travelling waves and conservation laws of a dimensional coupling system with korteweg-de vries equation, Applied Mathematics & Nonlinear Sciences, 2018, 3, 241-254.
[20] Naeem M., Siddiqui M.K., Guirao J.L.G., Gao W., New and modified eccentric indices of octagonal grid O_n^m, Applied Mathematics & Nonlinear Sciences, 2018, 3, 209-228.
[21] Pandey P.K., A new computational algorithm for the solution of second order initial value problems in ordinary differential equations, Applied Mathematics & Nonlinear Sciences, 2018, 3, 167-174.