Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform

Xian Zhong, Guang Yang, Lin Li, Luo Zhong

***Abstract*—With the development of the Internet, recommender systems have played a more and more important role in the field of big data processing, such as e-business. In order to deal with big data in recommender systems, we propose a clustering and correlation based collaborative filtering algorithm for cloud platform, which improves the traditional user-based collaborative filtering algorithm with k-medoids clustering and a data structure named correlation multi-tree in this paper. Firstly, we analyze the user-based collaborative filtering for cloud platform. On the basic of it, we propose a k-medoids based collaborative filtering algorithm for cloud platform by using the k-medoids clustering. It can solve the problem of data sparsity effectively. As a result, it can be more efficient with the recall rate and recommendation ratings improved at the same time. Considering the falling of recommendation accuracy by using clustering technology, this paper introduces a data structure named correlation multi-tree to correlate the user information and their neighbors information. It can be used to compute the extended user-item score, which makes full use of the correlation between data on cloud platform. As a result, the clustering and correlation based collaborative filtering algorithm for cloud platform proposed by us can improve the recommendation accuracy effectively, and ensure the effect of recommendation and the time efficiency at the same time. An extensive experimental evolution with Ali data sets on Hadoop cloud platform shows that our collaborative filtering algorithm has a better recommendation and is more efficient in handling big data.**

***Index Terms*—cloud platform, collaborative filtering, k-medoids clustering, correlation multi-tree, extended user-item score.**

I. Introduction

With the rapid development of Internet, the amount of data information generated by Internet is growing exponentially. Nowadays, due to the large amount of data base and its rapid growth, many industries, represented by electronic commerce, IT and telecom, have entered the era of big data[1][2]. Today, big data has become a great important problem in all walks of life, however the cloud computing can use a large number of computing nodes the which consisting of dynamically adjustment virtualized computing resources, and deals with the big data effectively through the paralleled and distributed computing technology[3]. In the era of big data, the data is not only with a big scale but also with a great value. It can create valuable derivatives for systems with large data, according to the data itself [4], [5]. Recommender system plays a very important role in value of the data aspect. How to efficiently use of big data of information system to process appropriate recommendation is a pressing problem[6]. Recommendation algorithm is the core of recommendation system.Recommendation algorithm tends to use the various data mining technologies including information retrieval, help users find their needed information from the jagged huge amounts of data, and avoid information overload caused by big data effectively [7], [8]. Therefore, the recommendation algorithm is given priority to collaborative filtering algorithm nowadays plays an important role in recommender system.

The main methods of the traditional collaborative filtering algorithm is user-based collaborative filtering which is based on user neighbors score to predict target users in scoring the designated goods [9]; But it may ignore the connection between the projects [10]. So, it is a common method that the recommendation item-based collaborative filtering is based on the project similarity. But in the face of sparse user behavior in e-commerce, recommendation accuracy is low, and recommendation effect is not ideal[11]. Using clustering techniques to divide objects into clusters of the clustering based collaborative filtering can improve the effect of recommendation effectively.

Clustering technology is a very effective means to solve the problem of data sparsity in the field of data mining[12]. Using clustering technology, user neighbors are divided into several clusters. It can be convenient for the choices of user neighbors, but also can find the user neighbors by one-time and calculate the neighbor users influence degree of the user projects easily[13]. In order to apply this technology, scholars have launched a deep research on it. Chen Ke-han aimed at the weak relationship between microblog social network. By combining the graph abstract method and based content similarity algorithm, he put forward the recommendation algorithm of a two-stage clustering GCCR, which is implemented based on user interest subject recommendation effectively and eased the data sparseness matrix problem[14]. Wei su-yun, aimed at the traditional collaborative filtering algorithm, proposes a project-based clustering of the global nearest neighbor collaborative filtering algorithm under the condition of the limitations of user rating data which is extreme sparse. This algorithm is according to the similarity between projects to clustering, and on this basis, calculates the user's local similarity, finally, given a method of using overlap degree factor, it adjusts the local similarity and describes the similarity between users more accurately[15]. However, using the clustering recommendation algorithm will cause a decline in accuracy, and is unable to meet many personalized requirements. Cao hong-jiang proposed an user clustering search algorithm combined with inverted index in the field of information retrieval and used "membership policy", reduces the time to calculate nearest neighbor. On the premise of guaranteeing recommendation accuracy effectively, it improves the scalability of collaborative filtering recommender system[16]. However, using the recommendation algorithm of clustering will cause a decline in accuracy, and is unable to meet many personalized requirements.

Xian Zhong, Guang Yang, Lin Li and Luo Zhong are with School of Computer Science & Technology, Wuhan University of Technology, 122 Luoshi Road, Wuhan 430070, China, e-mail: yangguangwhut@gmail.com.

Clustering technique is a very effective means in the field of data mining to solve the problem of data sparsity. In this paper, we will use the k-medoids clustering technology on the collaborative filtering based on clustering technology[12]. Using clustering technology, user neighbors are divided into several clusters. It not only can be convenient for the choices of user neighbors, but also can find the user neighbors by one-time, and it is convenient to calculate the neighbor users influence degree of the user projects[13]. Collaborative filtering on cloud platform, implement k-medoids clustering collaborative filtering algorithm algorithm for cloud platform. This algorithm can take the advantage of k-medoids clustering, solve the problem of data sparseness effectively and could generated recommended list more rapidly. On this basis, this paper puts forward an correlation multi-tree data structure which can correlate the user information and neighbor users into user information tuples. Thus, this algorithm can be able to generate more accurately extended user-item score reflecting the users project evaluation. Based on the k-medoids clustering for cloud platform and correlation multi-tree, we propose the clustering and correlation based collaborative filtering Algorithm for cloud platform. Using correlation multi-tree structure, we play the data value of users correlation, and overcome falling recommendation accuracy of the clustering technology. In ensuring the recommended time efficiency and recommend effect, we improve the accuracy and expansionary of the system. Through the analysis of experimental result, we can prove that the method is feasible.

II. Clustering and Correlation based Collaborative Filtering for Cloud Platform

A. Collaborative Filtering for Cloud Platform

1) Collaborative Filtering Process for Cloud Platform: The basic idea of collaborative filtering algorithm is according to the neighbor objects scores of user or project to predict target user ratings for goods with no score and recommend the predict score which the target users score on the highest number of goods[17]. General collaborative filtering algorithm generally has three steps: generating score matrix, choosing the most adjacent to the user (or project), generating the recommended product. This paper, represented based on user collaborative filtering, is referred to collaborative filtering. The specific steps are as follows:
Step 1: Calculating users behavior and acquiring user-project evaluation matrix, and converting to user-project evaluation of key-value pairs.
Step 2: Computing users similarity by using the Pearson correlation coefficient, and choosing the number of the most similar users as the nearest neighbor users.
Step 3: Predicting user ratings for goods with no score, according to the score of nearest neighbor users.
Step 4: Generating recommended list of items according to the predict items scores.

2) User-item score calculate: Information system of user behavior can be divided into two kinds, one kind is contain the user directly to score (such as film appreciation website) of the project, another kind is not to score the project directly (for example, the user shopping website). For the former, you can building user-score matrix directly by the user scores. And for the latter require user behavior to indirectly calculating user's interest in the project as a score[18]. General shopping e-commerce website user behavior including: click, buy, collect and into the shopping cart. We can calculate user-item score through these behaviors[19], and the computation formula is as follows:

$$S = N1 * W1 + N2 * W2 + N3 * W3 + N4 * W4 \quad (1)$$

Where, S represents user scores of the project, $N1$, $N2$, $N3$ and $N4$ represent the times of users' four behaviors which are click, buy, collect and go into the shopping cart, $W1$, $W2$, $W3$ and $W4$ are respectively corresponding to the weights of these behaviors.

3) User similarity calculating: We usually use the Pearson Correlation Coefficient to calculate the similarity between users in collaborative filtering recommendation system, and the user similarity computation formula is as follows:

$$Sim(a,b) = \frac{\sum_{p \in U_{a,b}} (r_{a,p} - \overline{r_a})(r_{b,p} - \overline{r_b})}{\sqrt{\sum_{p \in U_{a,b}} (r_{a,p} - \overline{r_a})^2} * \sqrt{\sum_{p \in U_{a,b}} (r_{b,p} - \overline{r_b})^2}} \quad (2)$$

Where, a and b represent the user, p represents the commodity; $U_{a,b}$ represents the intersection of a and b who has scored the commodity; $r_{a,p}$ represents the score which user a scores for goods p; $\overline{r_a}$ represents the arithmetic mean of all the related goods to user a. Calculation results $Sim(a,b)$ is the value of the fall in the interval [0, 1]. The bigger the $Sim(a,b)$ is, the bigger the similarity between user a and b is, on the other hand, the smaller the similarity between the user a and b is.

4) Recommend score calculating: Combined with the neighbor user scores and the user ratings which have been scored on items, we can get the score of items which haven't been scored, and the computation formula is as follows:

$$pred(u,i) = \overline{r_u} + \frac{\sum_{k \in U} [sim(u,k) * (r_k, i - \overline{r_k})]}{\sum_{k \in U} sim(u,k)} \quad (3)$$

Where $Pred(u,I)$ represents the recommending score of project marked by the user u. The users set U to represent user u's collection of nearest neighbors. $Sim(u,k)$ represents the similarity of user u and user k, where $r_{k,i}$ represents the score of the project I scored by user k, $\overline{r_u}$ and $\overline{r_k}$ respectively represent the scores that user u and user k score the arithmetic mean of all the relevant goods. Formula 3, on the basis of the weighted score based on similarity, we use the calculating method of the average normalized to eliminate the deviation which is brought by the different rating scales. By the above formula, we can predict target users rating for all items which have no score, and then predict the highest rated former Top-N items as recommended result feedback to the current user[20].

B. K-medoids based Collaborative Filtering Algorithm for Cloud Platform

1) K-means clustering and K-medoids clustering comparison: 1) K-means clustering and K-medoids clustering comparison: The k-means clustering algorithm and k-medoids clustering algorithm are the classical data mining algorithms. The k-medoids is more accurate at calculating the distance.

The k-medoids is aimed at overcoming the shortcomings of k-means that it is more sensitive to the noise and outlier data. Whats more, it will be more reasonable to use the k-medoids clustering to generate clusters with no restrictions of the size of cluster. The biggest difference between k-medoids and k-means is that k-medoids selects object from which each point is more closer in cluster as the cluster center point, while k-means chooses the center of gravity as the center point[21]. Therefore, with the comparison between the k-medoids clustering and k-means clustering, we can reduce the number of iterations, accelerate the gathering, and have a better clustering effect. Therefore, we propose k-medoids based collaborative filtering algorithm for cloud platform to cluster cloud platform data in this paper.

2) K-medoids based collaborative filtering process for cloud platform : First of all, according to the users and the rated items, we can cluster the user information. Then, we can deal with user-item score key-value pairs according to the user-based collaborative filtering. The concrete steps of k-medoids based collaborative filtering for cloud platform as follows:
Step 1: Calculating user-item scores matrix according to the user behavior and converting to user-item score key-value pairs.
Step 2: Using k-medoids clustering to divide the users into several clusters.
Step 3: The data processingdividing each cluster-unit into different node, calculating users similarity in clusters, then choosing the nearest neighbors.
Step 4: Calculating prediction score, generating the recommended list of items in each cluster.
Step 5: Reducing recommended list of different variety, generating the final recommended list of items.

3) K-medoids based collaborative filtering algorithm for cloud platform: The K-medoids Based Collaborative Filtering Algorithm for Cloud Platform can process user information and project information in the form of key-value pairs on cloud platform, and the same data nodes from the same cluster. The algorithm is described as follows:

In k-medoids clustering collaborative filtering algorithm for cloud platform(UDS), getItemnum (sMatrix) obtains user scoring items from each timestamp, sets this data set to be analysed, uses the k-medoids clustering to divide users into several clusters. It can effectively avoid the influence on sparse data of collaborative filtering, and the user similarity within each cluster is higher which can facilitate the efficient search in nearest neighbors. The algorithm, through the method of parallel reading the cluster data rather than reading all data, can generate recommended list quickly and have a higher time efficiency.

K-medoids clustering collaborative filtering algorithm for cloud platform is a kind of algorithm that uses clustering technology after the user is divided, then proceeds collaborative filtering. This algorithm can solve the problem of data sparsity. But in the process of dividing cluster, it doesnt make full use of the correlation among users which may lead to the decrease of the accuracy.

Algorithm 1 k-medoids based Collaborative Filtering algorithm for cloud platform

Input: user data sets(UDS)
Output: recommend item data sets(RDS)
  Procedure $kCF(RDS)$
  input UDS to RDS
  for $each \quad data \in UDS$ **do**
    $\langle user\_id, (item\_id, score, info)\rangle \leftarrow create\_keyvalue(data)$;
    $sMatrix \leftarrow \langle user\_id, (item_id, score, info)\rangle$;
  end for
  $numMatrix \leftarrow getItemnum(sMatrix)$;
  $Clusters \leftarrow kmedoids(sMatrix)$;
  for $each \quad data \in Clusters$ **do**
    $Scores \leftarrow Predict(user, neighbors)$;
    $\langle user, item\_list\rangle \leftarrow recom(user, Scores, sMatrix)$;
    $RDS \leftarrow \langle user, item\_list\rangle$;
  end for
  $RDS \leftarrow Reduce(RDS)$;
  return RDS
  EndProcedure

C. Clustering and Correlation based Collaborative Filtering for cloud platform

In order to reduce the influence on recommend accuracy of clustering, in this paper, we propose a correlation multi-tree structure, combining the neighbors user information and user information together to form user information tuples. In this way, we can make full use of the data on the user's relationship value and exert the value of big data for cloud platform. On the basis of the collaborative filtering algorithm based on cloud platform, k-medoids clustering and associated tree structure, we put forward the Clustering and Correlation based Collaborative Filtering for cloud platform(CCCF).

1) Correlation multi-tree: In the processing of big data information, especially in the electronic commerce system, users can be associated with projects through user behavior. Similarly, users who grade on the same project can be associated together through projects. To this end, we define a correlation multi-tree structure. In correlation multi-tree T, the root node r is a user, and other branch nodes are r's neighbor users. The leaf nodes are the users who have scored the projects. The weights of leaf nodes are the scores of projects. The following user project-score, for example, is to illustrate how the process of the construction of a tree. Users-project grades are shown in table I.

TABLE I: Users-Items score table

	$i1$	$i2$	$i3$	$i4$	$i5$
$u1$	9	8	0	0	8
$u2$	9	0	7	0	0
$u3$	0	9	0	9	6
$u4$	0	0	0	9	7
$u5$	0	0	5	6	0

We make the item score of items with no score to be 0, and defined that the two users which scored the same item are direct neighbor users. Here, $u1$ is the research object. As the table I shows, $u2$, $u3$, $u4$ and $u1$ are common grading

project. Therefore, $u2$, $u3$, $u4$ are the direct neighbor users of $u1$, they constitute the branch node of a correlation multi-tree in where $u1$ is the root node. The correlation multi-tree with $U1$ as a root node is shown in figure 1.

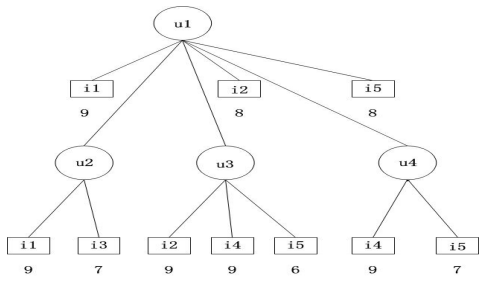

Fig. 1: The correlation multi-tree with $U1$ as a root node

Branch nodes($u2, u3, u4$) are the direct neighbor users of the root node $u1$. The leaf node is an item of terminal node corresponding to the users who have scored. The weight represents the user's rating on items (such as $u1$ of $i1$ scale of 9). If a user has multiple rating of the item, the weight is the average score.

2) Extended user-item score: Because the user correlation multi-tree can have more users, projects and its neighbor user rating associated directly, we can preliminary score and forecast to the user who has not scored the item(direct neighbors have been scored). Define the user rating on the propagation of the project calculation formula is:

$$score(u,i) = ru,i + \sum_{v \in Nu} (\frac{rv,i}{nu} * \frac{Iv}{Iu}) \qquad (4)$$

Where $r_{u,i}$ and $r_{v,i}$ represent the scores of user u and v scores on the item i respectively. N_u represents the collection of the direct neighbor user u. v is user u direct neighbor. n_u represents the number of u's direct neighbors which can be used to comprehensively reflect the multiple neighbor users score directly. I_u and I_v represent the number of user u and v who have scored items respectively, and they can be used to control the influence of different direct neighbor users (the more direct the neighbor rates items, the more trustworthy and influential it is). According to the formula 4, we can calculate the user u's extended user-item score:

$score(u1, i2) = 8 + 0 + \frac{9}{3} * \frac{3}{3} + 0 = 11$

$score(u1, i2) = 8 + 0 + \frac{9}{3} * \frac{3}{3} + 0 = 11$

$score(u1, i3) = 0 + \frac{7}{3} * \frac{2}{3} + 0 + 0 = 1.56$ $score(u1, i3) = 0 + \frac{7}{3} * \frac{2}{3} + 0 + 0 = 1.56$

$score(u1, i4) = 0 + 0 + \frac{9}{3} * \frac{3}{3} + \frac{9}{3} * \frac{2}{3} = 5$

$score(u1, i5) = 8 + 0 + \frac{6}{3} * \frac{3}{3} + \frac{7}{3} * \frac{2}{3} = 11.56$

3) Clustering and Correlation based Collaborative Filtering Process for Cloud Platform: Clustering and correlation based collaborative filtering for cloud platform, firstly establishes correlation multi-trees according to the user-score item matrix. Then according to the associated multi-trees, we compute extended user-item score, cluster the user information, and handle them with the user-based collaborative filtering. The specific steps of Clustering and correlation based collaborative filtering for cloud platform are as follows:

Step 1: According to the user's behavior, we compute and obtain user-item score score matrix and converted to key-value pairs of usersitem score.

Step 2: Putting each user as the root node, then building a correlation multi-tree.

Step 3: According to the correlation multi-trees, we compute the extended user-item score.

Step 4: Clustering the extended user-item score by using k-medoids, and then dividing the users into several clusters.

Step 5: Handling the data into different nodes as a unit, computing the similarity of users in the cluster, and selecting the nearest neighbor users.

Step 6: Computing the prediction score and producing a list of recommendation items of each cluster.

Step 7: Protocoling the recommendation lists of different clusters and producing the final list of recommendation items.

Algorithm 2 Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform

Input: user data sets(UDS)
Output: recommend item data sets(RDS)
Procedure $CCCF(UDS)$
input UDS to $HDFS$
for $each \quad data \in UDS$ **do**
$\langle user\_id, (item\_id, score, info)\rangle \leftarrow create\_keyvalue(data)$;
$sMatrix \leftarrow \langle user\_id, (item_id, score, info)\rangle$;
$num\_info \leftarrow count(data)$;
end for
for $each \quad user \in sMatrix$ **do**
$Corr\_Tree \leftarrow Create\_Tree(user, sMatrix)$;
for each $item \in user$ **do**
$Score(user, item) \leftarrow compute(item, Corr_Tree, num_info)$;
$s\_KeyValue \leftarrow Score(user, item)$;
end for
end for
$Clusters \leftarrow kmedoids(s\_KeyValue)$;
for $each \quad C \in Clusters$ **do**
$distances \leftarrow getDistance(C)$;
$neighbors \leftarrow chooseNeighbors(distance, C)$;
end for
for $each \quad user \in sMatrix$ **do**
$Scores \leftarrow Predict(user, neighbors)$;
$\langle user, item\_list\rangle \leftarrow recom(user, Scores, sMatrix)$;
$RDS \leftarrow \langle user, item\_list\rangle$;
end for
$RDS \leftarrow Reduce(RDS)$;
return RDS
EndProcedure

4) Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform: This algorithm uses the k-medoids (s_KeyValue) to cluster extended user-item score instead of using the k-medoids based Collaborative Filtering to cluster user-item score. This improved method not only can get more reasonable data clusters, but also it is actually a pretreatment for sparse initial data, which can reduce the number of iterations and speed up the clustering.

Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform kCCF(UDS), proposes a data structure named correlation multi-tree on the basis of a k-medoids based Collaborative Filtering algorithm, which associates user's information with its direct neighbor. Ac-

cording to the correlation between users, the algorithm can effectively play the value of data. It not only can ensure the recommendation precision, but also owns the advantages in synergistic speed of clustering and correlation based collaborative filtering algorithm for cloud platform at the same time.

III. Experimental results and analysis

A. The experimental data and configuration

This paper uses Tmall's users' real behavior log provided by the race of Ali's big data and from which we extract various user's behavior like clicking on the items, purchasing, collecting, putting the items into the shopping cart. There are 4 moths data of Tmall's users' behavior, including about 1000 users, thousands of brands and a total of 182881 of the users' behavior information of Tmall. Users and brands are respectively made to a certain degree of data sampling and all users' behaviors are accurated to the level of the day.

In this paper, the experiments are carried out in the Hadoop cloud platform, which uses a NameNode and 9 DataNode cloud clusters built by 10 machines. The experimental configuration of each experiment node is on the VMware virtual machine, and the version is Ubuntu operating system installed on 12.04.3.VM. To achieve the related algorithms, we use the JDK of the 1.7.0_21 version, the Hadoop of the 1.1.2 version. The related configuration as shown in table II:

B. The experimental index

1) Running Time: Running time can be used to measure the efficiency of recommendation algorithms. There exists parallelism advantage in cloud platform which can improve the serial algorithm in the aspect of operational efficiency. The calculation formula of the total running time is as follows:

$$T = T_i + T_u + T_p + T_{other} \tag{5}$$

Where T is the total running time. T_i represents the calculating time of computing all users-item score, T_u represents the time of computing the similarity and selecting the nearest neighbor, T_p represents computing time of the prediction score, T_{other} represents other time overhead (cloud platform is mainly the time of piecewise aggregating and data linking). The shorter the operating time, the higher the efficiency of the recommendation algorithm, on the contrary, the lower the efficiency described.

2) Recommendation effect:

a) Accuracy: Accuracy is one of the most commonly used evaluation index of general recommendation algorithm, and it is mainly used to measure the precision of user's Purchasing behavior predicted by recommendation algorithm. Its formula is as follows[22]:

$$Pre = \sum_{j=1}^{T} h_j / \sum_{i=1}^{T} b_i \tag{6}$$

b) Recall rate: Recall rate is another the most commonly used measurement in recommendation algorithm, and it is mainly used to measure the credibility of user's purchase behavior predicted by recommendation algorithm. Its formula is as follows[22]:

$$\mathrm{Re} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{H_i} h_j}{\sum_{i=1}^{N} \sum_{j=1}^{B_i} b_j} \tag{7}$$

c) Recommendation rating: When evaluating whether a recommendation algorithm is good or not, often requires to consider the precision(accuracy) and reliability(recall); for the same algorithm, the hit rate and recall rate often tend to exhibit negative correlation[23] when parameters are in the process of changing. Therefore, we comprehensively evaluate recommendation effect of recommendation algorithm by using score computing function of SPR. SPR function is as follows:

$$SPR = \frac{2 * \mathrm{Pr}\, e * \mathrm{Pr}\, e}{(\mathrm{Pr}\, e + \mathrm{Re})} \tag{8}$$

Where, Pre represents the precision of recommendation algorithm and Re represents the recall rate of recommendation algorithm. SPR puts the precision and the recall rate into a unified index, comprehensively considering the user's demand for the precision and the recall rate of products recommended by system, and the higher the value, the better the recommendation effect and more able to satisfy the user's actual demand for products.

C. The Experimental results

1) Running Time: According to Tmall's transaction records related to data sets provided by the race of Ali's big data, on which we replicate data, we obtain different data sizes of user-related information data and statistic the running time of the collaborative filtering algorithm of different data sets. The experimental results are shown in the following table III

The experimental results show that, User-based collaborative filtering is more efficient than Item-based collaborative filtering for this data set. Therefore, we set user-based collaborative filtering as the foundation of other collaborative filtering. When the data reaches a certain size, if we don't use collaborative filtering for cloud platform, we can't handle the data. Comparing collaborative filtering for cloud platform with collaborative filtering without using clustering, collaborative filtering that using clustering technique is higher in time efficiency, because it can effectively solve the problem of data sparsity. With the expansion of data size, the advantages of running speed will be more obvious. Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform, not only has the advantage of using clustering technique, but also can preprocess data according to its correlation multi-tree structure, which has higher operational efficiency and has the superiority of time.

2) Recommendation effect: In this paper, based on user-based collaborative filtering, we collaboratively filtered the data provided by the race of Ali's big data by four methods that are collaborative filtering algorithm (CF algorithm), collaborative filtering algorithm for cloud platform (Cloud CF algorithm), a k-medoids based Collaborative Filtering algorithm (kCF algorithm) and Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform (CCCF algorithm) on the basis of user collaborative filtering.

TABLE II: The configuration of the distributed platform

	CPU	Memory	Hard disk	Operating System	Runtime environment
NameNode	i5-2410M	8G	750GB	Ubuntu 12.04	Hadoop-1.1.2 VMware-workstation 8
DataNode	i5-2310M	4G	500GB	Ubuntu 12.04	Hadoop-1.1.2 VMware-workstation 8

TABLE III: Comparison Table of Running time

size of data set	Item-based collaborative filtering	User-based collaborative filtering	collaborative filtering for cloud platform	k-medoids based collaborative filtering for cloud platform	Clustering and Correlation based collaborative filtering for cloud platform
0.5MB	19s176s	19s731ms	18s421s	17s058ms	18s203ms
5MB	1min584s	59s383ms	17s497s	15s776ms	13s955ms
50MB	7min93s	5min502s	59s461ms	45s177ms	37s676ms
500MB	$>$ 10h	$>$ 10h	11min919s	3min325s	2min56s
5GB	Unable to process	Unable to process	49min571s	24min147s	17min914s

When computing these collaborative filtering algorithms in the recommending of different numbers of items, it returns the precision and recall rate. The experimental results are the average of each collaborative filtering algorithm which has been done many times and the recommendation ratings is computed by computing function of SPR(units of precision, recall rate and recommendation ratings are %). The experimental results as shown in the following figure:

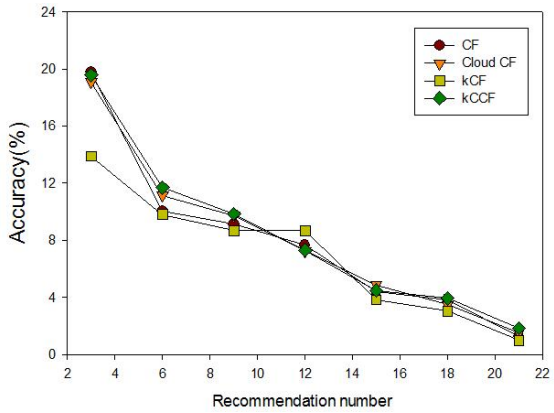

Fig. 2: Comparison chart of recommendation precision

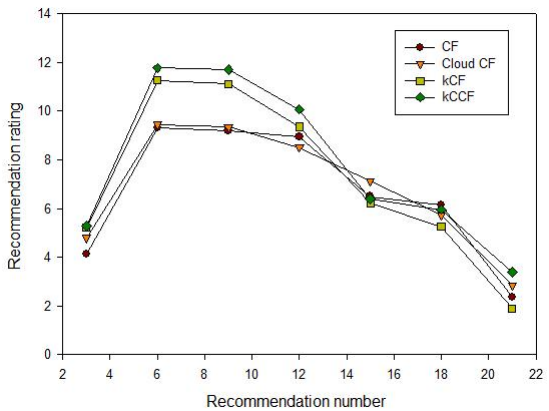

Fig. 3: Comparison chart of recommendation recall rate

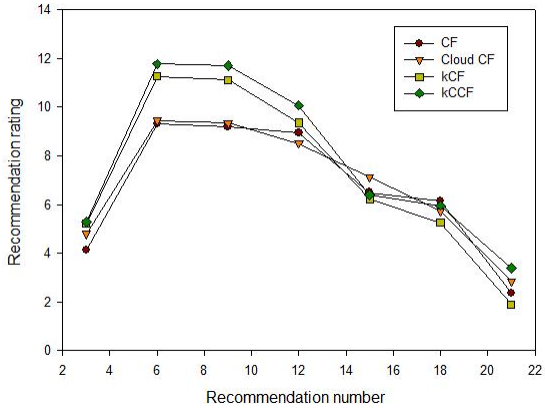

Fig. 4: Comparison chart of recommendation ratings

The experiments show that recommendation number of data sets ranges from 6 to 9 can get the highest score and the best recommendation effect. Comparing precision, recall rate and recommendation ratings of these three algorithms whose recommendation numbers are ranged from 6 to 9, we can find that a k-medoids based Collaborative Filtering algorithm (kCF algorithm) which uses clustering technology to overcome data sparsity and Clustering and Correlation based Collaborative Filtering Algorithm for Cloud Platform (kCCF algorithm) which can significantly improve recall rate (average increased by 5.75%), can effectively improve the recommendation effect (score) (average increased by 1.79%). However, kCF algorithm will to cause a decline of precision in a certain degree. KCCF algorithm can comprehensively take the interaction relationship between users into consideration, and can effectively play the data value of user's and its direct neighbor's information, while ensuring recommendation effect and operation efficiency. It also guarantees the recommendation precision.

IV. Conclusion

In order to overcome data sparsity which collaborative filtering for cloud platform can not effectively solve, especially data sparsity in the background of big data, the

paper proposed a k-medoids based Collaborative Filtering algorithm for cloud algorithm on the basis of k-medoids. Using recommendation system of this algorithm can overcome data sparsity in a certain degree, produce recommendation lists rapidly and improve the recommendation effect. On the basis of k-medoids clustering based Collaborative Filtering algorithm for cloud platform and correlation multi-tree, the paper proposed clustering and correlation based collaborative filtering for cloud platform, which overcomes the negative impact of the clustering technique to recommendation precision in a certain degree. While ensuring time efficiency and recommendation effect, it also can improve recommendation precision and it is a kind of an efficient recommendation algorithm which is suitable for recommendation system of processing big data. In the future, we will do a further research which can play more value of data and design a recommendation system of higher recommendation with a better operation efficiency.

REFERENCES

[1] J.-Z. Li and X.-M. Liu, "An important aspect of big data: *Data usability," Journal of Computer Research and Development*, vol. 50, no. 1, p.6, 2013.

[2] B. P. Lester, "Operator fusion in a data parallel library." *IAENG International Journal of Computer Science*, vol. 39, no. 1, pp50-63, 2012.

[3] Q. Li and X. Zheng, "The present reseach of cloud computing," *Computer Science*, vol. 38, no. 4, pp. 32–37, 2011.

[4] P. Lv, L. Zhong, D. Cai, and Y. Wu, "Chinese commentary effectiveness mining product characteristics based on crf," *Computer engineering and science*, pp. 359–366, 02 2013.

[5] P. Lv, L. Zhong, and K. Tang, "Research on comprehensive evaluation of customer satisfaction of online product reviews," *Chinese Journal of Electronics*, pp. 740–746, 04 2012.

[6] L. Guo, J. Ma, and Z.-M. Chen, "A social recommendation algorithm combining correlated relationship between recommendation objects," *Chinese Journal of Computers*, vol. 37, no. 1, pp. 219–228, 2014.

[7] J. Bobadilla, F. Ortega, A. Hernando, and Á. Arroyo, "A balanced memory-based collaborative filtering similarity measure," *International Journal of Intelligent Systems*, vol. 27, no. 10, pp. 939–946, 2012.

[8] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," *Knowledge-Based Systems*, vol. 37, pp. 146–153, 2013.

[9] Z.-D. Zhao and M.-S. Shang, "User-based collaborative-filtering recommendation algorithms on hadoop," in *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*, pp. 478–481.

[10] L. Guo, J. Ma, Z. Chen, and H. Jiang, "Learning to recommend with social relation ensemble," in *Proceedings of the 21st ACM international conference on Information and knowledge management 2012*, pp. 2599–2602.

[11] Q.-L. Ba, X.-Y. Li, and Z.-Y. Bai, "Clustering collaborative filtering recommendation system based on svd algorithm," in *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on*, pp. 963–967.

[12] K.-H. Tang, L. Zhong, L. Li, and G. Yang, "Urban tunnel clean up the dirty data based on clara optimization clustering," *Journal of Applied Sciences*, vol. 11, pp. 1980–1984, 2013.

[13] M. C. Pham, Y.-W. Cao, R. Klamma, and M. Jarke, "A clustering approach for collaborative filtering recommendation using social network analysis," *J. UCS*, vol. 17, no. 4, pp. 583–604, 2011.

[14] K.-H. Chen, P.-P. Han, and J. Wu, "Heterogeneous social network recommendation algorithm based on user clustering," *Chinese Journal of computers*, vol. 36, no. 2, pp. 349–359, 2013.

[15] S.-Y. Wei, N. Ye, J. Zhu, and et al., "The global nearest neighbor clustering collaborative filtering algorithm based on item clustering," *Journal of computer science*, vol. 39, no. 12, pp. 149–152, 2014.

[16] H.-J. Cao and K. Fu, "Clustering search method research in collaborative filtering recommendation system," *Computer engineering and applications*, vol. 50, no. 5, pp. 16–20, 2014.

[17] M. Esfahani and F. Alhan, "New hybrid recommendation system based on c-means clustering method," *Information and Knowledge Technology*, no. 5, pp. 145–149, 2013.

[18] L. Q, "Enhancing collaborative filtering by user interest expansion via personalized ranking," *IEEE Trans Syst Man Cybern B Cybern*, vol. 42, no. 1, pp. 218–233, 2012.

[19] L. Li and X. Chen, "Extraction and analysis of chinese microblog topics from sina," *Cloud and Green Computing (CGC), 2012 Second International Conference on*, vol. 90, no. 1, pp. 571–577, 2012.

[20] W. L, E. Chen, L. Q, and et al, "Leveraging tagging for neighborhood-aware probabilistic matrix factorization," in *Proceedings of the 21st ACM international conference on Information and knowledge management 2010*, pp. 1854–1858.

[21] X. Zhang, K. Gong, and G. Zhao, "Parallel k-medoids algorithm based on mapreduce," *Journal of Computer Applications*, vol. 33, no. 4, pp. 1023–1025, 2013.

[22] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2010.

[23] X. Chen, L. Li, G. Xu, Z. Yang, and M. Kitsuregawa, "Recommending related microblogs: A comparison between topic and wordnet based approaches." in *AAAI 2013*, pp. 2417–2418.