Future Generation Computer Systems 93 (2019) 1046–1054

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Category Preferred Canopy–K-means based Collaborative Filtering algorithm

Jianjiang Li [a], Kai Zhang [a], Xiaolei Yang [a], Peng Wei [a], Jie Wang [a], Karan Mitra [b], Rajiv Ranjan [c,*]

[a] Department of Computer Science and Technology, University of Science and Technology, Beijing, China
[b] Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Sweden
[c] School of Computer Science, China University of Geosciences, Wuhan, China

HIGHLIGHTS

- CPCKCF algorithm is proposed to deal with data sparsity and scalability.
- CPCKCF algorithm reduces data dimensionality and simplifies similarity calculation.
- CPCKCF algorithm improves recommendation accuracy and instantaneity.

ARTICLE INFO

Article history:
Received 3 May 2017
Received in revised form 13 October 2017
Accepted 7 April 2018
Available online 25 May 2018

Keywords:
Recommender system
Collaborative Filtering
Data mining
Category preferred ratio

ABSTRACT

It is the era of information explosion and overload. The recommender systems can help people quickly get the expected information when facing the enormous data flood. Therefore, researchers in both industry and academia are also paying more attention to this area. The Collaborative Filtering Algorithm (CF) is one of the most widely used algorithms in recommender systems. However, it has difficulty in dealing with the problems of sparsity and scalability of data. This paper presents Category Preferred Canopy–K-means based Collaborative Filtering Algorithm (CPCKCF) to solve the challenges of sparsity and scalability of data. In particular, CPCKCF proposes the definition of the User–Item Category Preferred Ratio (UICPR), and use it to compute the UICPR matrix. The results can be applied to cluster the user data and find the nearest users to obtain prediction ratings. Our experimentation results performed using the MovieLens data set demonstrates that compared with traditional user-based Collaborative Filtering algorithm, the proposed CPCKCF algorithm proposed in this paper improved computational efficiency and recommendation accuracy by 2.81%.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

With the development of information communication technologies and the Internet, the data is increasing at an exponential scale [1–4]. It is incredibly challenging for the information consumers to gain valuable and useful information from massive amounts of data; while for information provider, it is also challenging to provide remarkable content. Therefore, we have witnessed the emergence of a vast number of search engines and recommender systems. The users can find interesting information by inputting the keywords into the recommender systems or search engines. However, the users cannot expect to get interesting results when they cannot enter precise keywords into the search engine. The recommender systems, on the other hand, may provide interesting information in this case. The recommender systems obtain the information that meets the needs and interests of users through the method of analyzing the users' past behavior, and by excavating their preferences and building the model of their interests.

At present, the techniques recommender systems are using include association rules, content-based recommendation, Collaborative Filtering and hybrid approach. As a traditional algorithm of the recommender systems, the Collaborative Filtering algorithm is the most well known and accepted algorithm due to its many advantages. For instance, there is no need to consider the content of recommended items; it provides a serendipitous recommendation for the users. Further, it is easy to implement with low data dependency and offers accurate recommendation results.

* Corresponding author.
E-mail address: rranjans@gmail.com (R. Ranjan).

https://doi.org/10.1016/j.future.2018.04.025
0167-739X/© 2018 Elsevier B.V. All rights reserved.

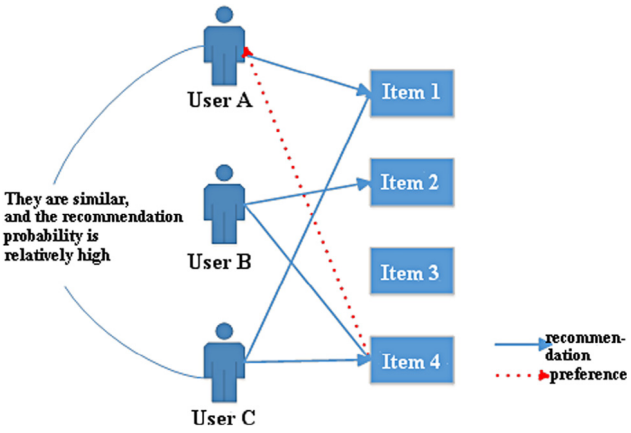

Fig. 1. The principle of User Based Collaborative Filtering algorithm.

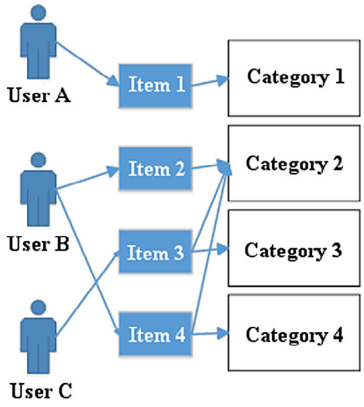

Fig. 2. The relations of users, items and category preferred ratio.

However, the Collaborative Filtering algorithm also has several limitations such as low scalability when dealing with large amounts of data, and the problem of a cold start. Further, the traditional Collaborative Filtering algorithm needs to compute the similarities of increasing number users to all other users, and it requires higher computation efficiency. It is a significant challenge to improve computation speed for an online recommender system. Also, the number of users and items is vast; however, most users just rate a small part of items, so the data used to calculate similarities between users and items is sparse. Finally, it comes to the condition that the recommendation results may not be satisfactory.

This paper proposes Category Preferred Canopy–K-means based Collaborative Filtering (CPCKCF) algorithm that addresses the challenges mentioned above and optimizes recommender systems regarding computation performance and prediction accuracy. Section 2 introduces the traditional User-based Collaborative Filtering Algorithm (UCF), propose the concept of User–Item Category Preferred Ratio (UICPR) and the implementation details of the CPCKCF with clustering. Section 3 presents the results related to the CPCKCF algorithm. Results validation is via theoretical and experimental analysis. Section 4 presents the related work in the area of Collaborative Filtering. Finally, Section 5 presents the conclusion and future work.

2. Category preferred Canopy–K-means based Collaborative Filtering algorithm

User-based Collaborative Filtering algorithm is one of the earliest recommendation algorithms. It is widely accepted because of its many advantages mentioned before. This paper designs a new recommendation algorithm with the method of utilizing the idea of category preference to cluster the data and optimizes recommender systems in term of prediction accuracy and instantaneity.

2.1. Traditional User-based Collaborative Filtering algorithm

User-based Collaborative Filtering (UCF) algorithm was proposed in 1992 and applied successfully in mail filtering systems then in news filtering by research institutions GroupLens in 1994. It is one of the most widely used algorithms in the domain of recommender systems until 2000. The algorithm collects the data of users preference, then uses KNN algorithm to calculate the cluster of the nearest users and concludes the common preference of N nearest users, it finally recommends non-common preference to users based on the degree of common preference. The principle of the algorithm is illustrated in Fig. 1.

From Fig. 1, assuming that user A likes item 1 and item 3, user B likes item 2 and item 4, user C likes item 1 and item 4. User A is the objective user, user A and user C have same preference in item 4, but user A and user B do not have common preference. Considering that user C and user A have higher similarity, their preference are more close to each other. And user C also likes item 1 which user A has not used before. Thus, it is a good idea to recommend item 1 rather than item 2 or 3 to user A. The UCF algorithm can be concluded into 3 steps:

(1) Calculate the similarities of all users to the objective user.
(2) Choose N users that have TOP-N similarities to objective user and obtain the collection of the nearest users.
(3) Employ the weighted mean values of the nearest users to predict the objective user's rating.

The UCF algorithm discovers non-common preference data in user set through calculating the nearest users set which has similar common preference and provides non-common preference data to objective user.

2.2. The improved User-based Collaborative Filtering algorithm

This paper proposes the Category Preferred Canopy–K-means based Collaborative Filtering Algorithm (CPCKCF) based on the UCF. The improved algorithm reduces the complexity of computation and increases the accuracy of recommendation results as well. The CPCKCF will be demonstrated in detail as follows.

2.2.1. User–Item Category Preferred Ratio

The data of recommender systems is excessively large and sparse, so it is necessary to reduce the dimension of the rating matrix. The CPCKCF puts forward the concept of User–Item Category Preferred Ratio (UICPR) and makes the rating data converge to the preferred ratio of item category. This paper will illustrate the concept of UICPR from three aspects (i.e. principle, computation and advantages) respectively.

(1) **Principle:** Generally speaking, every item has one or more category attributes. There is a big difference in preferred ratio of users between different category items. The rating data of films can be taken as an example, some users grade mainly on romantic films, but others prefer to grade on science fiction movies. As shown in Fig. 2, item 2 and item 4 that user B like belong to the category 2. Thus, the users who have the same category preferred

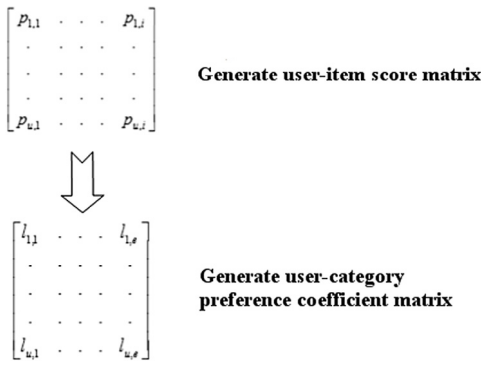

Fig. 3. Calculations of category preferred ratio.

ratio can be found out by defining preferred ratio of users to different items and it will improve the efficiency of recommender system.

(2) **Computation:** The category preferred ratio of user to a certain type of items can be demonstrated from two parts: (i) The proportion of user's ratings to a certain type of items in all his/her scores; and (ii) The proportion of his/her rating items in all items. If the value of the former is larger, it shows that the number of ratings and the scores are higher, which indicates that the user may like this certain kind of item more. And the latter value is the weight of the item. Different kinds of items distribute unevenly and the algorithm does weighted penalty to excessively popular category to some extent.

Firstly searching the users' rating records and then calculating the matrix of category preferred coefficient. The process of computation is illustrated as Fig. 3, $p_{u,i}$ is the score of user u to item i, $l_{u,e}$ is the category preferred coefficient of user u to category e.

(3) **Advantages:** As mentioned above, the dimension of user–item category preferred coefficient matrix is smaller than that of original user–item score matrix. And the recommendation accuracy will be improved by 2.81% according to the experimental results.

2.2.2. Data clustering

(1) **The principle of cluster:** Traditional UCF algorithm has some drawbacks, so this paper proposes a clustering algorithm which has been applied in data mining. The algorithm divides users into some parts according to the characteristic of cluster algorithm and searches the nearest users in a certain cluster based on the distance between the objective user and the center of a cluster. Finally, it calculates the similarities and prediction scores. The modified algorithm can improve computation efficiency and instantaneity of the system.

(2) **The implementation of the algorithm:** This paper employs widely used K-means algorithm to cluster users' data. Considering the selection problem of value k in K-means algorithm, the Canopy and K-means will be utilized sequentially in the clustering.

2.2.3. Category Preferred Canopy–K-means based Collaborative Filtering algorithm (CPCKCF)

The steps of CPCKCF are as follows.

(1) Calculate the UICPR with the data of user–item rating scores.
(2) Cluster users' data according to the UICPR.
(3) Compute the distance between objective user and all cluster centers and find out the nearest users in the nearest cluster.
(4) Calculate the similarities of the nearest users to objective user.
(5) Predict the scores according to the data in (4).

The main difference between CPCKCF and UCF can be stated in three aspects:

(1) The CPCKCF proposes the concept of UICPR, and obtains the category preferred ratio matrix which can reduce the sparsity of data by calculating the related coefficient.
(2) The CPCKCF clusters the data of users with Canopy and K-means algorithm, and simplifies the subsequent calculation. The process of CPCKCF and UCF are illustrated in Fig. 4 where bold and black boxes in the CPCKCF indicate the better processes compared with the UCF.

The detailed steps of CPCKCF will be stated respectively.

(1) **Calculate the UICPR**: Assuming that the total category of items is E, each category of item is e, the preferred coefficient of user u with category e is $l_{u,e}$, the definition of users' category preferred ratio is as follows:

$$l_{u,e} = \frac{\sum_{i \in Z(e)} p_{u,i}}{\sum p_u} \cdot \log \frac{|d(E)|}{|d(e)|} \tag{1}$$

where $\sum_{i \in Z(e)} p_{u,i}$ is the total scores of user u rate on each category e, $\sum p_u$ is the total scores of user u on all categories of items, $|d(E)|$ is the number of all items and $|d(e)|$ is the number of items in category e. $\frac{\sum_{i \in Z(e)} p_{u,i}}{\sum p_u}$ is the rating proportion of user u in category e. If the rating proportion is higher, it shows that user u pays more attention to this category of items, so the subsequent recommendation should focus on this category. $\log \frac{|d(E)|}{|d(e)|}$ is the category e and that of all items, it penalizes the excessively popular category. Finally the UICPR matrix L shown in Fig. 5 can be obtained by traversing user–item score matrix.

(2) **Clustering user data:** K-means algorithm is the typical algorithm in the domain of data mining. However, the number of k may have great influence in cluster results. Thus, the CPCKCF utilizes Canopy algorithm as a prepositive algorithm and use the output of Canopy algorithm as the input of K-means algorithm, which can ensure the stability of cluster results.

The CPCKCF sets two thresholds θ_1 and θ_2, calculates the distance between category preferred ratio vector $\overrightarrow{l_u}$ and initial points, and divides vector into *canopy*1 or *canopy*2 according to the value of distance. Then the CPCKCF utilizes the output of Canopy as the input of K-means algorithm, computes the distance of each vector and cluster center, finally updates all cluster center iteratively until cluster centers are invariable.

(3) **Search the nearest users**: The CPCKCF calculates the distance between objective user and cluster centers after clustering the matrix Q'. If the distance is lower, it shows that similarities of objective user and the users in the cluster will be higher. Thus, the CPCKCF selects N points that have the highest similarities as the nearest users. The process is illustrated in Fig. 6. Different values of N have different impacts on recommendation accuracy. The calculation formula of similarity is as follows:

$$w_e(u, u1) = \frac{\sum_{e \in E}(p_{u,e} - \overline{p_u}) \cdot (p_{u1,e} - \overline{p_{u1}})}{\sqrt{\sum_{e \in E}(p_{u,e} - \overline{p_u})^2} \cdot \sqrt{\sum_{e \in E}(p_{u1,e} - \overline{p_{u1}})^2}} \tag{2}$$

where $w_e(u, u1)$ is the category preferred similarity of user u and $u1$, E is the category of item, $p_{u,e}$ is score of user u rate on category e, $p_{u1,e}$ is score of user $u1$ rate on category e, $\overline{p_u}$ is the mean score of user u rate on all categories of items, and $\overline{p_{u1}}$ is the mean score of user $u1$ rate on all categories of items.

(4) **Calculate the fitting similarity:** The similarities after selecting the TOP-N nearest users are computed. The traditional formula

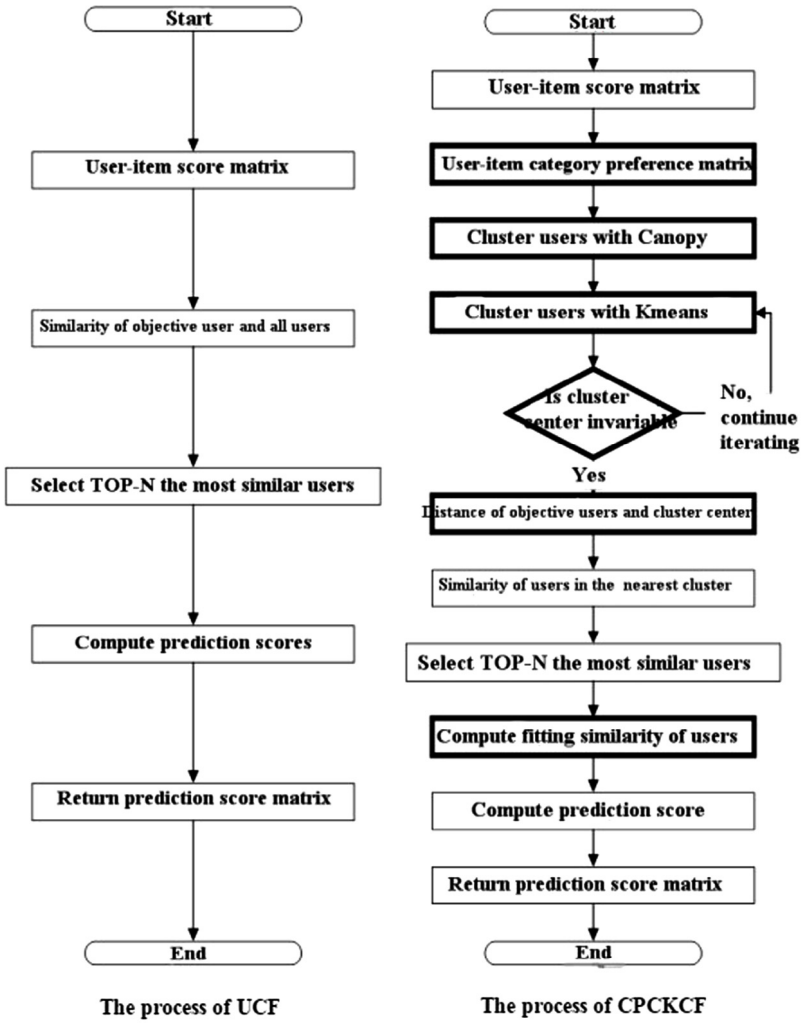

Fig. 4. Comparison of the process of CPCKCF and the process of UCF.

$$\begin{bmatrix} l_{1,1} & \cdots & l_{1,e} \\ \vdots & \cdots & \vdots \\ l_{u,1} & \cdots & l_{u,e} \end{bmatrix}$$

Fig. 5. Matrix of User–Item Category Preferred Ratio.

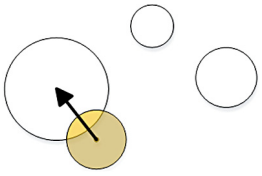

Fig. 6. Objective user selects the most nearest users.

of similarity is as follows:

$$w_u(u, u1) = \frac{\sum_{i\in I}(p_{u,i} - \overline{p_u}) \cdot (p_{u1,i} - \overline{p_{u1}})}{\sqrt{\sum_{i\in I}(p_{u,i} - \overline{p_u})^2} \cdot \sqrt{\sum_{i\in I}(p_{u1,i} - \overline{p_{u1}})^2}} \quad (3)$$

The calculation of traditional similarity utilizes user–item score matrix, however, because the set of the nearest users has been computed according to the step of (3), users' similarities with UICPR matrix can be also computed.

Therefore, this paper uses linear fitting method to coalesce the two similarities and acquires better calculation results. The improved formula is as follows:

$$w(u1, u2) = w_u(u1, u2) \cdot \chi + w_e(u1, u2) \cdot (1 - \chi) \quad (4)$$

where $w(u1, u2)$ is the fitting similarity of $u1$ and $u2$, $w_u(u1, u2)$ is the traditional similarity, $w_e(u1, u2)$ is the user category preferred similarity, and χ is the fitting parameter.

The range of χ is [0,1] , the weight of two similarities can be balanced by adjusting the parameters. The impact of fitting similarity on recommendation results will be discussed in the third part.

(5) **Predict the scores:** The algorithm calculates the prediction scores of objective user to item i after obtaining users' fitting similarities according to formula 5 and acquires the prediction score matrix Q' which is shown as Fig. 7.

$$p'_{u,i} = \frac{\sum_{u1\in Z(N)} w(u, u1) \times p_{u1,j}}{\sum_{u1\in Z(N)} |w(u, u1)|} \quad (5)$$

$$\begin{bmatrix} p_{1,1}' & \cdots & p_{1,i}' \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ p_{u,1}' & \cdots & p_{u,i}' \end{bmatrix}$$

Fig. 7. Matrix of users–items prediction scores.

where $p'_{u,i}$ is the prediction score of user u to item i, $w(u1, u)$ is the fitting similarity of $u1$ and u , $Z(N)$ is the set of TOP-N users in the cluster.

The CPCKCF is demonstrated as Algorithm 1.

2.3. The analysis of CPCKCF

The advantages of CPCKCF are mainly on three aspects.

(1) **Increase the recommendation accuracy:** The algorithm imports the concept of UICPR and utilizes it to cluster users. The two similarities are linearly fitted when the users' similarities are calculated and it can increase the recommendation accuracy.
(2) **Improve the calculation efficiency:** The algorithm imports cluster algorithm and decreases the computational complexity of users' similarities. Thus, the algorithm improves computation efficiency and enhances the instantaneity of the recommender system.
(3) **Enhance the robustness:** The CPCKCF utilizes Canopy algorithm as a prepositive algorithm and reduces the impact of value k on K-means algorithm. It makes the algorithm become more robust.

Algorithm 1 CPCKCF algorithm

```
1: Initialize Q
2: for each u, i
3:     for each e
4:         Use formula 1 to calculate UICPR l_{u,e}
5:     end for
6: end for
7: Generate UICPR matrix L
8: Initialize θ1 and θ2
9: for each u, e
10:    Calculate the distance of θ1 and θ2 and
11:    Join canopy1 or canopy2 according to the standard
12: end for
13: for each u, e in Canopy
14:    for each center
15:        Use formula 3 to calculate the similarities of cluster center w_{u1,u2}
16:    end for
17:    The objective user selects N nearest users
18: end for
19: for each center
20:    Update the cluster centers
21: end for
22: if satisfy the condition of convergence
23:        go to line 25
24: else   go to line 13
25: for each u
26:    Calculate the distance f between the objective user u and the cluster center
27: end for
28: if min(f) ∈ cluster C
29:    Add u into cluster C
30: end if
31: for each u in cluster C
32:    Use formula 2 to calculate user category preferred similarity w_e
33: end for
34: Select N highest values in w_e as the nearest users
35: Use formula 3 to calculate the traditional similarity w_u
36: Use formula 4 to calculate the fitting similarity w
37: Calculate prediction scores
38: return Q'
```

3. Results analysis

This section highlights the experimental results of CPCKCF, and verifies the theoretical analysis with experimental results.

3.1. Experimental data and setup

There are many open test data sets of recommender system, such as MovieLens, Netflix, etc. MovieLens is one of the most accepted data sets in academia. So this paper selects it as the source of test data. MovieLens is mainly composed of three parts: scores, videos and links. Different data sets contain the information of film scores which is rated by different users, ranging from 0.5 to 5 points. To decrease the sparsity of data, it should be ensured that each user has rated at least 20 films. In addition, evaluation time, film classifications, film tags and link information like IMDB and TMDB are relatively sufficient in MovieLens as well. Table 1 illustrates some basic information of MovieLens data set.

According to the instantaneity of experimental data and the scale of the algorithm, this paper exploits 10M data as the experimental standard. A PC can satisfy experiment requirements as we use offline experiment. Table 2 illustrates the experimental environment.

The experiment of recommender systems can be mainly divided into two parts (i.e. offline and online experiments). The online experiments need a lot of real-time feedback of online users, so it is difficult to accomplish the experiments. This paper will use the method of offline experiments. The data set is divided into training set (80%) and test set (20%).

(1) **Accuracy:** Accuracy is the most common and direct recommendation standard of recommender systems. This paper will use the RMSE as the accuracy standard. And the formula of RMSE is as follows:

$$RMSE = \frac{\sqrt{\sum_{i=1}^{n}(|p_i' - p_i|)^2}}{n} \tag{6}$$

where p_i is the real score of item i, p_i' is the prediction score of item i, n is the number of scores. The recommendation accuracy can be measured through difference of real score and prediction score.

(2) **Instantaneity:** The data that needs to be processed by the recommender algorithm is usually quite large, which indicates that the algorithm also needs a high standard of instantaneity. This paper exploits two indexes to measure the recommendation effect of algorithm.

(a) Running time of recommendation: The algorithm optimizes the time of the whole process and each step of recommendation, compared with the traditional algorithm, this paper uses the running time to measure the instantaneity of recommender system.

(b) Search rate of the nearest users: This paper refers to the definition in [5] and utilizes the search rate of the nearest users (SR) as evaluation standard of instantaneity.

The nearest neighbor coincidence degree $r1$ indicates that the nearest users searched by CPCKCF is coincident with UCF. And the formula of the coincidence degree of the nearest users $r1$ is as follows:

$$r1 = \frac{|Z_{UCF}(u) \bigcap Z_{CPCKCF}(u)|}{|Z_{UCF}(u)|} \tag{7}$$

Table 1
Data set of MovieLens.

Size of data	Number of users	Number of films	Number of ratings	Tags
0.1M	943	1682	100000	0
1M	6040	3952	1000209	0
10M	71567	10681	10000054	95580
20M	138493	27278	20000263	465564

Table 2
Experimental environment.

Operation system	CPU	Memory
Windows 7	Intel core i7 2630QM	8 GB

Table 3
Parameter meaning of CPCKCF.

Parameters	Meanings	Values
χ	The parameter of fitting similarity	0.6
N	The number of the nearest users	30

where $Z_{UCF}(u)$ is the nearest users' set of user u when searching by UCF, $Z_{CPCKCF}(u)$ is the nearest users' set of user u when searching by CPCKCF.

The range ratio of the nearest users $r2$ indicates the proportion of the nearest users searched by CPCKCF to all users. The formula of $r2$ is as follows:

$$r2 = \frac{|Z_{CPCKCF}(u)|}{|U|} \tag{8}$$

where U is the set of all users, $Z_{CPCKCF}(u)$ is the set of users who are in the same cluster with user u.

The search rate of the nearest users is defined with the ratio of $r1$ and $r2$.

$$SR = \frac{r1}{r2} = \frac{|Z_{UCF}(u) \bigcap Z_{CPCKCF}(u)| \cdot |U|}{|Z_{UCF}(u)| \cdot |Z_{CPCKCF}(u)|} \tag{9}$$

3.2. Experimental results and analysis of CPCKCF

This paper will calculate the user–item score matrix firstly and set the related parameters of the algorithm after initializing user–item score matrix Q. The meaning of related parameters is shown as Table 3:

This paper has adjusted the parameters with the method of cross validation in order to prevent the model from being over-fitting. The values of better experimental results are shown in Table 3. The impact of related parameters on the recommendation results will be discussed below.

(1) Accuracy: The experiment uses the data in MovieLens and RMSE as the accuracy standard. This paper will verify the impact of the number N of the nearest users and fitting similarity parameter χ on recommendation accuracy respectively through the experiments.

(a) The impact of the number of the nearest users on recommendation accuracy: Firstly, this paper will set the value of χ to be 0.4, set N to be the independent variables and set CPCKCF and UCF to be the dependent variable. The experimental results are illustrated as Fig. 8.

The values of RMSE in two algorithms both decrease firstly but increase later when the value N increases. In the beginning, the number of the nearest users is too small, and the samples of data are excessive sparse, so a few bad points have some impacts on the whole data. However, when the number of the nearest users is extremely large, the data integrity increases, however, the long tail noise data will reduce the recommendation effect.

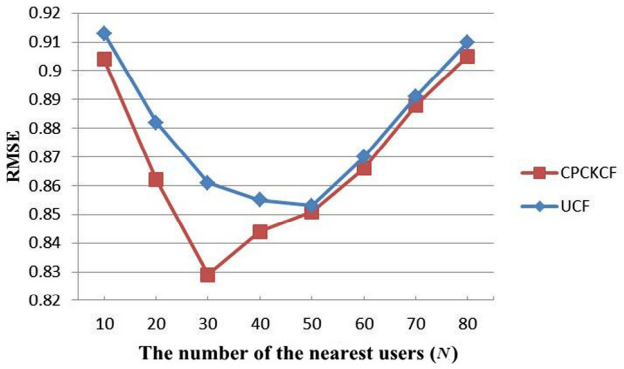

Fig. 8. The impact of the number of the nearest users on RMSE.

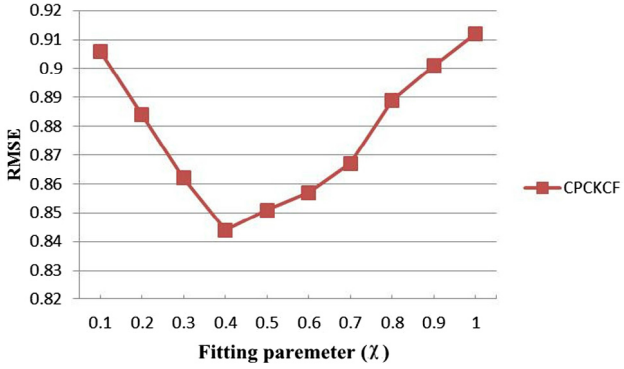

Fig. 9. The influence of the fitting parameters on RMSE.

Compared with UCF, the extreme value of CPCKCF appears at the point of N=30, RMSE=0.829 and that of UCF appears at the point of N=50, RMSE=0.853. CPCKCF utilizes the data of UICPR and selects the nearest users after clustering. Thus CPCKCF can achieve a better result in a smaller range.

The value of RMSE of CPCKCF always smaller than that of UCF and the extreme values of two algorithms differ by 2.81% ((0.853–0.829)/0.853). The result indicates that CPCKCF which clusters with UICPR is better in recommendation accuracy.

(b) The impact of fitting similarity on accuracy: This paper sets the number of the nearest users to be 40, sets χ as the independent variables and finds out its relationship with the value of RMSE as Fig. 9.

When the fitting parameter is 0.4, recommender system has the best recommendation results. The dimension of $w_u(u1, u2)$ is too large and that of $w_e(u1, u2)$ is too small, so the similarities calculated by $w_u(u1, u2)$ and $w_e(u1, u2)$ respectively both have some difference.

(2) Instantaneity: According to the principle of CPCKCF, this algorithm optimizes the instantaneity mainly in the way that it can acquire high recommendation accuracy even if searching in a small domain of the nearest users. This paper uses SR to measure the instantaneity of CPCKCF.

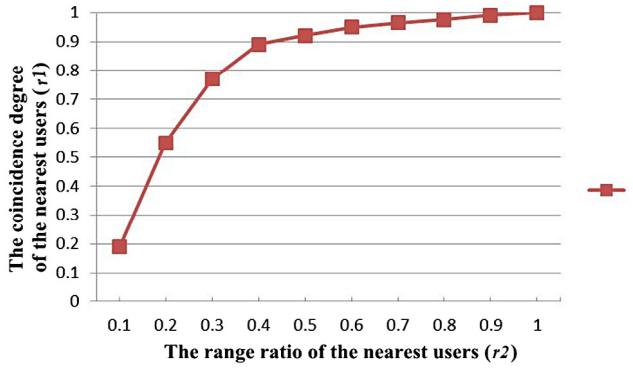

Fig. 10. Relationship of nearest neighbors range ratio and the nearest neighbors coverage.

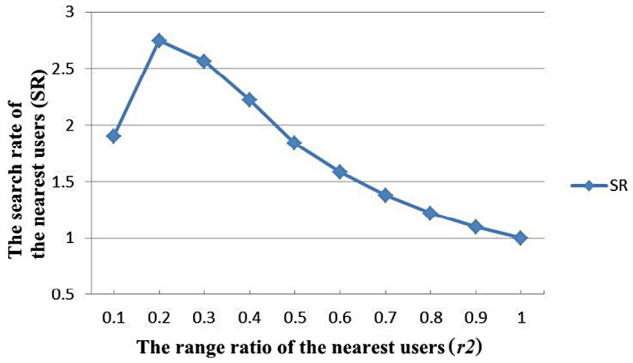

Fig. 11. Relationship of nearest neighbors range ratio and the nearest neighbors search rate.

Because the original data is excessively large, this paper selects 10% data randomly as test sample and sets the number of the nearest users to be 40 according to the definition of formula 9, then calculates the value of SR. The relationship of $r1$ and $r2$ is illustrated as Fig. 10.

When the range ratio of the nearest users (i.e. $r2$) is 30%, the coincidence degree of the nearest users (i.e. $r1$) is 77% and when the $r2$ is 40%, $r1$ achieves 89%. As shown in Fig. 11, SR is always bigger than 1 and achieves the extreme value when $r2$ is equal to 0.2. Thus, it can be concluded that CPCKCF searches in a small domain of the nearest users and obtains better recommendation results compared with UCF. The CPCKCF reduces a significant amount of calculation of similarities and improves a lot in instantaneity of a recommender system.

4. Related work

Yu et al. [6] point that it is possible to select a subset of user profiles (Profile Space), and search the nearest neighbor of the objective user in Profile Space in order to improve the speed of the recommendation algorithm. Luo et al. [7] present incremental Collaborative Filtering algorithm based on regularized matrix factorization and design two updating mechanism to update the rating data timely. Shani [8] uses the Markov model to analyze the users' historical data and recommends the items to users from the analysis, and it has a significant development in recommendation accuracy. Raghavan S [9] et al. employ mass fraction to increase or decrease the weight of single user's scores and improve the recommendation accuracy.

George et al. [10] employ Bregman co-clustering algorithm to cluster the users and items simultaneously, the algorithm finds the cluster which objective users and items are in and calculates the prediction scores of objective users and items. To deal with the problems of scalability and sparseness of the user profiles, Zhou et al. [11] describe a modified CF algorithm called alternating-least-squares with weighted-λ-regularization (ALS-WR), and the performance of ALS-WR improves with both the number of features and that of ALS iterations. The reference [12] proposes a novel typicality-based Collaborative Filtering (TCF) recommendation method which imports the idea of object typicality from cognitive psychology. TCF finds "neighbors" of users based on user typicality degrees in user groups, and it has higher recommendation accuracy and lower time cost than other CF algorithms.

Xiang [13] presents interesting research about the impact of time behavior on recommender systems and builds the users' preference model for the recommendation tasks of score prediction and Top-N. Sun [14] et al. present a modeling method of user's timing behavior and coalesce the nearest users' set into Collaborative Filtering algorithm based on probability matrix decomposition. With the development of big data computing framework like MapReduce, it becomes a trend to design and run Collaborative Filtering algorithm on distributed computing framework. Ref. [15] improves the User-based Collaborative Filtering algorithm by normalization method, and the algorithm can be run on the MapReduce on the Hadoop platform, greatly improves the recommendation accuracy and computational efficiency. To deal with the efficiency of Matrix Factorization based Collaborative Filtering (MFCF) recommendation, Yang et al. [16] re-implement MFCF algorithm on the platform of MapReduce and propose a four-step process of MFCF, each of which is sent to be treated as a MapReduce task.

Ref. [17] proposes a recursive prediction algorithm which eases the problem of a sparse matrix and improves the recommendation accuracy. The algorithm makes the nearest users to join the forecasting process even though they have not graded for the given items, and for users whose scores are uncertain, predicts its recursion. Gupta et al. [18] propose a framework that prediction using item based Collaborative Filtering is combined with prediction using demographics based user clusters in an adaptive weighted scheme. Wu et al. [19] consider a hybrid approach that combines content-based approach with Collaborative Filtering called co-clustering with augmented matrices (CCAM), which is based on information-theoretic co-clustering but further considers augmented data matrices like user profile and item description.

Acilar et al. [20] present a Collaborative Filtering model based on Artificial Immune Network and use the algorithm of Artificial Immune Network to condense the rating matrix, as a result, the number of users in the matrix and the sparsity of rating data can be decreased. To deal with the problem of cold start, Massa et al. [21] propose to transmit the trust over the trust network to find users that can be trusted by the active user, and items appreciated by these trustworthy users can be recommended to the active user.

Wei et al. [22] propose two recommendation models to solve the complete cold start (CCS) and incomplete cold start (ICS) problems for the new items, and the models are based on a framework of tightly coupled CF algorithm and deep learning neural network. Russell et al. [5] present Discrete Wavelet Transformation based Collaborative Filtering algorithm, which uses wavelet to compression data space, and makes the number of item rating vector reducing by times. Finally, the algorithm recommends the items to users by traditional Collaborative Filtering in reduced data space. For the condition of data sparsity in recommender systems, Liu et al. [23] present the collaborative filtering algorithm based on "star users".

5. Conclusion and future work

People's demand for personalized information is becoming stronger with information overload. Accordingly, the importance of the recommender system is increasingly highlighted. However, the traditional Collaborative Filtering algorithm has some drawbacks regarding sparsity of data, undesirable instantaneity, and scalability issues with large amounts of data. This paper proposes Category Preferred Canopy–K-means based Collaborative Filtering algorithm. The algorithm reduces the dimension of data by computing User–item Category Preferred Ratio and clusters the users in the meanwhile. It has also simplified the selection of the nearest users and the calculation of similarities. The experimental results show that CPCKCF proposed in this paper is better than commonly used UCF algorithm in both recommendation accuracy and instantaneity.

The future work include tuning the algorithm based on online experiment and trying other clustering algorithms such as Mixture-of-Gaussian clustering to optimize CF algorithm.

Acknowledgment

This work was supported in part by Beijing Key Subject Development Project, China (XK10080537).

References

[1] T.T. Hills, T. Noguchi, M. Gibbert, Information overload or search-amplified risk? Set size and order effects on decisions from experience, Psychon. Bull. & Rev. (2013) 1023–1031.
[2] Lizhe Wang, Weijing Song, Peng Liu, Link the remote sensing big data to the image features via wavelet transformation, Cluster Comput. 19 (2) (2016) 793–810.
[3] Weijing Song, Peng Liu, Lizhe Wang, Sparse representation-based correlation analysis of non-stationary spatiotemporal big data, Int. J. Digit. Earth 9 (9) (2016) 892–913.
[4] Dan Chen, Yangyang Hu, Lizhe Wang, Albert Y. Zomaya, Xiaoli Li, H-PARAFAC: Hierarchical parallel factor analysis of multidimensional big data, IEEE Trans. Parallel Distrib. Syst. 28 (4) (2017) 1091–1104.
[5] S. Russell, V. Yoon, Applications of wavelet data reduction in a recommender system, Expert Syst. Appl. (2008) 2316–2325.
[6] K. Yu, A. Schwaighofer, V. Tresp, et al., Probabilistic memory-based collaborative filtering, IEEE Trans. Knowl. Data Eng. (2004) 56–69.
[7] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowl.-Based Syst. (2012) 271–280.
[8] G. Shani, R.I. Brafman, D. Heckerman, An MDP-based recommender system, in: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc, 2002, pp. 453–460.
[9] S. Raghavan, S. Gunasekar, J. Ghosh, Review quality aware collaborative filtering, in: ACM Conference on Recommender Systems, ACM, 2012, pp. 123–130.
[10] T. George, S. Merugu, A scalable collaborative filtering framework based on co-clustering, in: Data Mining Fifth IEEE International Conference On, IEEE, 2005.
[11] Y. Zhou, D. Wilkinson, R. Schreiber, et al., Large-scale parallel collaborative filtering for the netflix prize, in: International Conference on Algorithmic Applications in Management, Springer, Berlin Heidelberg, 2008, pp. 337–348.
[12] Y. Cai, H. Leung, Q. Li, et al., Typicality-based collaborative filtering recommendation, IEEE Trans. Knowl. Data Eng. (2014) 766–779.
[13] L. Xiang, Research on the key technology of dynamic recommender system, Autom. Chin. Acad. Sci. (2012).
[14] G. Sun, L. Wu, Temporal behavior based collaborative filtering recommendation algorithm, J. Softw. (2013) 2721–2733.
[15] J. Dong, Y. Qin, X.Y. Sun, et al., Research on improved collaborative filtering recommendation algorithm on MapReduce, in: MATEC Web of Conferences, EDP Sciences, 2016.
[16] X. Yang, P. Liu, Collaborative Filtering Recommendation using Matrix Factorization: A MapReduce Implementation, 2014.
[17] J.Y. Zhang, P. Pearl, A recursive prediction algorithm for collabo-rative filtering recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems, ACM, 2007, pp. 57–64.
[18] J. Gupta, J. Gadge, A framework for a recommendation system based on collaborative filtering and demographics, in: Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on, IEEE, 2014, pp. 300–304.
[19] M.L. Wu, C.H. Chang, R.Z. Liu, Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices, Expert Syst. Appl. (2014) 2754–2761.
[20] A.M. Acilar, A. Arslan, A collaborative filtering method based on artificial immune network, Expert Syst. Appl. (2009) 8324–8332.
[21] P. Massa, P. Avesani, Trust metrics in recommender systems, in: Computing with Social Trust, Springer, London, 2009, pp. 259–285.
[22] J. Wei, J. He, K. Chen, et al., Collaborative filtering and deep learning based recommendation system for cold start items, Expert Syst. Appl. (2017) 29–39.
[23] Q. Liu, The Research of Key Algorithm in Collaborative Filtering Recommendation System, Zhejiang University, 2013.

Jianjiang Li is currently an associate professor at University of Science and Technology Beijing, China. He received his Ph.D. degree in computer science from Tsinghua University in 2005. He was a visiting scholar at Temple University from Jan. 2014 to Jan. 2015. His current research interests include parallel computing, cloud computing, parallel compilation and big data.

Kai Zhang is currently a master degree candidate in University of Science and Technology Beijing, China. He received his B.S. Degree in automation from University of Science and Technology Beijing in 2016. His current research interests include parallel computing, cloud computing, and social network applications.

Xiaolei Yang has received his master degree from University of Science and Technology Beijing, China. His current research interests include parallel computing, cloud computing and recommender systems.

Peng Wei is currently a master degree candidate in University of Science and Technology Beijing, China. He received his B.S. Degree in computer science and technology from Qufu Normal University in 2016. His current research interests include parallel computing, cloud computing and big data.

Jie Wang is currently a master degree candidate in University of Science and Technology Beijing, China. She received her B.S. Degree in computer science and technology from Tangshan Normal University in 2015. Her current research interests include cloud computing and parallel computing.

Karan Mitra is an Assistant Professor at Luleå University of Technology, Sweden. He received his Dual-badge Ph.D. from Monash University, Australia and Luleå University of Technology in 2013. He received his MIT (MT) and a PGradDipDigComm from Monash University in 2008 and 2006, respectively. He received his BIS (Hons.) from Guru Gobind Singh Indraprastha University, Delhi, India in 2004. His research interests include quality of experience modeling and prediction, context-aware computing, cloud computing and mobile and pervasive computing systems. From January 2012 to December 2013 he worked as a researcher at CSIRO, Canberra, Australia. He is a member of the IEEE and ACM.

Dr. Rajiv Ranjan is a Reader in the School of Computing Science at Newcastle University, UK; chair professor in the School of Computer, Chinese University of Geoscience, Wuhan, China; and a visiting scientist at Data61, CSIRO, Australia. His research interests include grid computing, peer-to-peer networks, cloud computing, Internet of Things, and big data analytics. He has published about 200 research papers (including 120+ journal papers). His papers have received 7770+ Google Scholar citations in total, he has an h-index and i10-index of 36 and 74 respectively. His papers have also received 1700+ citations and h-index of 16; according to Thomson Reuters Journal Citation Report (goo.gl/mJVphW). He also has an Scopus (Author ID:22980683700) h-index of 19 and total citations >2900. Ranjan has a Ph.D. in computer science and software engineering from the University of Melbourne (2009). Contact him at raj.ranjan@ncl.ac.uk or http://rajivranjan.net.