FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 40 (2015) No. 2
ISSN 0867-6356
e-ISSN 2300-3405
DE GRUYTER OPEN
DOI: 10.1515/fcds-2015-0010

DATA SENSITIVE RECOMMENDATION BASED ON COMMUNITY DETECTION

Chang SU, Yue YU, Xianzhong XIE, Yukun WANG

Abstract. Collaborative filtering is one of the most successful and widely used recommendation systems. A hybrid collaborative filtering method called data sensitive recommendation based on community detection (DSRCD) is proposed as a solution to cold start and data sparsity problems in CF. Data sensitive similarity is combined with Pearson similarity to calculate the similarity between users. α is the control parameter. A predicted rating mechanism is used to solve data sparsity problem and to obtain more accurate recommendation. Both user-user similarity and item-item similarity are considered in predicted rating mechanism. β is the control parameter. Moreover, in the constructed K-nearest neighbour set, both user-community similarity and user-user similarity are considered. The target user is either in the community or has some correlation to the community. Calculating the user-community similarity can cope with cold start problem. To calculate the recommendation, movielens data sets are used in the experiments. First, parameters α and β are tested and DSRCD is compared with traditional collaborative filtering recommendation algorithm (TCF) and Zhao's algorithm. DSRCD always has better results than TCF. When K = 30, we have better performance results than Zhao's algorithm.

Keywords: Community detection, Collaborative filtering algorithm, Cold start, Predicted rating mechanism

1. Introduction

With the rapid development of Web 2.0, the Internet has become interactive allowing users not only obtain information but also share information, i.e., shopping experience, item ratings, product reviews, etc. Large-scaled information is generated such as users' interests, opinions, ratings, etc., which are useful to understand the preferences of users. Because of

*C. SU, Y. YU, X. XIE and Y.WANG are affiliated with the School of Computer Science and Technology, at Chongqing University of Posts and Telecommunications, Chongqing, China. X. XIE, the corresponding author, is also affiliated with Chongqing Key Lab of Mobile Communications Technology at Chongqing University of Posts and Telecommunications, Chongqing, China. Our email addresses are changsu@cqupt.edu.cn, 974834832@qq.com, xiexzh@cqupt.edu.cn, airfer@qq.com.

the complexity of vast amounts of information, buyers may find it difficult to sort through the mass number of products, and merchants have difficulty knowing customer needs based on their purchasing records and product rating scores. Traditional search engines such as Google, Baidu, 360 search, etc., can provide information retrieval service. With the same key word, all the users will obtain the same retrieval results from search engines, but not personalized service. How to recommend an appropriate product to a particular user is of great interest to merchants and researchers.

Personalized recommendation takes the advantage of the users' preference information such as user's personal interests, online shopping habits, products rating score information and makes personalized recommendation for users. Many personalized recommendation systems have been widely used in various fields such as B2C, movies, music. In addition, there are many famous recommendation systems in the field of e-commerce, such as Amazon, YouTube, Taobao, Jingdong, Dangdang and in movie field such as DouBan, MovieLens. Recommendation algorithms play an important role in the accuracy of the recommendation systems.

Collaborative filtering is one of the most successful and widely used and implemented recommendation algorithms. The assumption of collaborative filtering is that if user i has the same opinion on issue x with user j, then there is a high probability they would have the same opinion on another issue y. Collaborative filtering usually has three phases: calculating the similarity between users or items; forming neighbourhood by finding K similar users or items; finding the top N items based on ratings of users in the neighbourhood.

As collaborative filtering methods make the recommendations based on users' rating history. The new user has to rate a sufficient number of items to enable the system to provide precise recommendation. Otherwise, the system cannot make the recommendations. This limitation is called the cold start problem. There are other challenges for CF, e.g., data sparsity, scalability, grey sheep, etc. Therefore, the studies of personalized recommendation systems, especially in the context of social networks, both from a theoretical point of view and a practical point of view are importance [2][3][8][13][15].

In this paper, we aim to show in some respects how to improve the performance of collaborative filtering recommendation. We propose a hybrid collaborative filtering model called data sensitive recommendation based on community detection (DSRCD). We summarize our main contributions or strong points as follows:

1. We propose a new similarity calculation method called 'data sensitive similarity' which considers the arithmetic difference between two users' rating information. It is combined with Pearson similarity to calculate similarity between users.
2. We propose a new predicted rating mechanism to solve the data sparsity problem and to have more accurate recommendation. We use both user-user similarity and item-item similarity to predict the rating.
3. We use a community detection method to cope with the cold start. When we construct the K-nearest neighbor set, we consider not only user-community similarity but also user-user similarity.

This paper is organized as follows. Section 2 introduces some related work in collaborative filtering recommendations. Section 3 presents the data sensitive recommendation algorithm based on community detection. Section 4 reports simulation results. Concluding remarks and future directions are presented in Section 5.

2. Related work

Personalized recommendation algorithms are divided into four categories, including content-based recommendation algorithms, association-rules-based recommendation algorithms, collaborative filtering recommendation algorithm, and hybrid recommendation algorithm. Collaborative filtering algorithms have been widely used and have been very successful. Collaborative filtering algorithms are divided to three main categories: the memory-based collaborative filtering, model-based collaborative filtering and hybrid collaborative filtering.

In memory-based collaborative filtering algorithms, much related research [1] [16] [17] has been done to improve Pearson correlation or cosine similarity calculation. According to the principal of the algorithms, memory-based collaborative filtering algorithms can be divided into user-based memory algorithms and item-based memory algorithms. Sarwar et al [16] first proposed a method which utilizing a user-score matrix and users' similarity to make the recommendation. Shih et al [17] proposed a collaborative filtering algorithm based on user similarity calculation in 2005. Adomavicius G et al [1] presented a way to reverse the user to study the frequency of a collaborative filtering algorithm approach. In 2013, Zhao QQ et al [24] proposed a memory-based collaborative filtering algorithm via propagation. The algorithm based on similarity propagation models corrected similarity degree calculating between user-user and item-item in order to generate a more reasonable set of nearest neighbours. They utilized the two aspects of the information to complete the recommendation process.

The idea of model-based collaborative filtering algorithms is to use the existing data for statistical analysis, mathematical modelling and the user's behaviour model to predict the user's preference. One of the biggest differences between memory-based collaborative filtering algorithm and Model-based collaborative filtering algorithm is whether user's behaviour model is used to make recommendations. More model-based recommended models include the Bayesian model proposed by Breese et al [5] in 1998, the probability class correlation model proposed Getoor et al [10] in 1999, the maximum entropy model proposed by Pavlov [15] in 2002 etc. Sun G.F. et al in [18] proposed a collaborative filtering recommendation algorithm based on sequential behaviour. This method captured the sequential behaviour of users and products so that a more accurate neighbourhood can be found. Zhang Y et al in [25] proposed an autonomy-oriented personalized tag recommendation algorithm, which used a latent Dirichlet allocation like probabilistic approach. It modelled user's preference information on tag and provided autonomy oriented personalized tag recommendation. Because of the changing number of users and the increasing of user-score, score data sets are constantly changing. Therefore, user behaviour model created according to relevant data should be updated every once in a while, and in the training of new user behaviour models also consume a lot of time. Hence most of model-based collaborative filtering algorithms are applicable to fewer users' interest changes and slow data updating speed.

Hybrid collaborative filtering which combined memory-based model and model-based model overcomes the limitation of native CF algorithms. In hybrid recommendation algorithms collaborative filtering is combined with other recommendation algorithms. Balabanović M [6] et al proposed a hybrid recommendation system which is based on the capacity of collaborative filtering algorithms. Users' similarity is calculated based on the configuration files, rather than on the rating information of the item in order to overcome the sparseness. Good N et al [11] proposed a similarity calculation method through different filters (filter bots). They used a special kind of agent content analysis as a supplement of collaborative filtering. Melville P, et al [12] added bonus points for the user's score vector through the method based on text analysis in the collaborative filtering system. User information with higher bonus points will have priority for recommendation. Yoshii K et al [22] combined collaborative filtering algorithm and audio analysis technology for music recommendations. Girardi and Marinho [9] used domain ontology technology in the collaborative filtering system for the Web recommendation.

Today, the boundaries between different disciplines have become relatively vague. Using the knowledge of other disciplines to solve problems in the field of personalized recommendation has become a trend. For example, some collaborative filtering algorithms combined the social network, community detection and traditional collaborative filtering algorithm to improve recommendation accuracy and its performance. Related research includes A Collaborative Filtering Method using Topological-Potential Based Community Discovery Strategy, proposed by Chen [7] et al, Research on Personalized Recommendation Algorithm Based on Social Network, proposed by Zhu et al [23], Leveraging Overlapping Communities Detection Improve Personalized Recommendation in Folksonomy Networks, proposed by Su et al [19]. This paper presents also research technology about how to community detection to mitigate problems such as data sparsity, cold start and other issues. Section four presents how to use the community detection to make accurate recommendations.

3. Data Sensitive Recommendation Algorithm

3.1. Construct User-user Networks

The user-item network is converted to a user-user network in order to make the recommendations among users. The user-item network is represented in matrix R, in which R_{ij} represents the rating that user i scores item j. The range of the rating value is [1, z], where z is usually set to 5 or 10,because not everyone gives his rating to the items and the users score is only a small portion of all items; therefore, the matrix R is a sparse matrix.

If there are two users and their scores are similar, then it can be inferred that they may have similar preferences for products, therefore, the similarity of the users is calculated and stored in matrix U, where U_{ij} represents the similarity between user i and user j. The user-user network is constructed in which the nodes are users and the edges are similarities between users. There are methods to calculate the similarity such as cosine similarity, and Pearson similarity.

3.1.1. Cosine similarity

Cosine similarity can calculate the similarity between users, but it does not take data sensitivity into consideration. In an extreme case, there are two vectors $\vec{X}(1,1)$ and $\vec{Y}(5,5)$, where 1 represents a negative rating and 5 represents a positive rating. Through calculation it can be found that the cosine similarity of the two vectors is large, which means the rating of two users are very similar. While the rating vectors of two users varies greatly. In this case, the results of the cosine similarity do not match the real situation.

3.1.2. Pearson similarity.

Pearson similarity has much in common with cosine similarity, which does not take data sensitivity into consideration. For example, there are two vectors $\vec{X}(1,2,3,2,1)$ and $\vec{X}(3,3,4,5,4)$, where the vector $\vec{X}$ represents some low rating of selected items; vector $\vec{Y}$ represents some high rating of the items. Although the two vectors show a great difference, the Pearson similarity of the two vectors is 1, which means the two vectors are almost the same.

Therefore, data sensitivity similarity is defined in Eq. (1) based on Pearson similarity. $R_{\max}$ represents the maximum rating value that a user can score. In Eq. (2), the Pearson similarity represented as $sim_{Pearson}$. $\overline{R}_{ui}$, $\overline{R}_{uj}$ show the average rating value of user i and user j, respectively. Eq. (1) and Eq. (2) are combined to calculate the similarity of user i and user j in Eq(3), where α is the control parameter.

$$sim_{seni}\left(u_i,\mathrm{u}_j\right)=\left(1-\sqrt{\frac{\sum_{t\in I_{ui}\cap I_{uj}}\left(R_{u_it}-R_{u_jt}\right)^2}{\sum_{t\in I_{ui}\cap I_{uj}}\mathrm{R}_{\max}{}^2}}\right) \quad (1)$$

$$Sim_{Pearson}\left(u_i,\mathrm{u}_j\right)=\frac{\sum_{t\in I_{u_i}\cap I_{u_j}}\left(R_{u_it}-\overline{R_{u_i}}\right)\left(R_{u_jt}-\overline{R_{u_j}}\right)}{\sqrt{\sum_{t\in I_{ui}\cap I_{uj}}\left(R_{u_it}-\overline{R_{u_i}}\right)^2}\sqrt{\sum_{t\in I_{ui}\cap I_{uj}}\left(R_{u_jt}-\overline{R_{u_j}}\right)^2}} \quad (2)$$

$$Sim\left(u_i,u_j\right)=\left(1-\alpha\right)\times sim_{Pearson}\left(u_i,u_j\right)+\alpha\times sim_{senti}\left(u_i,u_j\right) \quad (3)$$

3.2. Constructing Nearest Neighbour Set based on Community Detection

The aim of the community detection [14] is to find some groups, the entities in which have many properties in common. If the entity is a user, then the users in the same group may have the same interests for some items. Therefore, the community detection method can be used to construct the nearest neighbour set. In this paper, the algorithm proposed by Blondel et al [4] is used for community detection.

In traditional collaborative filtering algorithm based on users, the construction of the nearest neighbour set uses the similarity of users. First, the similarities of users are sorted in descending order according to similarity to the target user. In the similarity sorting list, the top K users are selected. In the data sensitive recommendation based on community detection (DSRCD), community detection is first used to find the groups with the same interests. When constructing the nearest neighbour set, the users in the same groups are considered in the first place, which not only improves the recommendation accuracy but also decreases the cold start problem existing in traditional collaborative filtering algorithm. If a user scores some items, then the user belongs to some groups according to certain rules.

3.2.1. Predicted rating mechanism

It has been shown that not all users score items. In real recommendation systems, the items that users score only account for a small part of the number of items. In this subsection, a predicted rating method for items which are missing rating information is proposed, which decreases the influence of data sparsity that causes recommendation inaccuracy.

Suppose there are five users: User1, User2, User3, User4, User5, and five items: Item1, Item2, Item3, Item4, Item5. The ratings information can be seen in Table 1. The symbol '?' represents that that item has no rating information. When the algorithm needs the rating information of item2 that user3 scores, or need the rating information of item1 that user4 scores, there is no rating information about these items; therefore, a predicted rating strategy is needed.

Table 1. User-Item rating example

Items Users	Item1	Item2	Item3	Item4	Item5
User1	2	1	1	2	1
User2	3	1	4	3	1
User3	1	?	2	2	2
User4	?	3	2	1	3
User5	5	3	3	5	2

Through observing item1 and item4, it can be found that the ratings information of the two items are similar, the rating of item1 that user4 scores may be 1 or 2. Similarly, the rating of item2 that user3 scores may be 2 or 1 based on the rating information between item2 and item5; therefore, the missing rating can be predicted by the ratings of the similar items.

Given $\vec{X}(x_1, x_2, ..., x_n)$, x_i represents the rating information of item X that user i scores. Given $\vec{Y}(y_1, y_2, ..., y_n)$, y_i represents the rating information of item y that user i scores, Rmax represents the maximum rating value that a user scores. The similarity calculation equation of the items can be seen in Eq. 4.

$$sim_{item}(X,Y) = (1-\alpha) \times sim_{Pearson}(X,Y) + \alpha \times sim_{senti}(X,Y) \tag{4}$$

$Sim_{senti}(X,Y)$ and $sim_{Pearson}(X,Y)$ can be calculated using Eq. (1) and Eq. (2), respectively. The nearest items set Neigh(Ix) can be constructed using $sim_{item}(X,Y)$. The value of parameter α can refer to Eq. (3). After community detection, a user belongs to a community or a few communities; the users in the same community may have much common in scoring; and the range of ratings may be high, such as (3, 5) or may be low such as (1, 3). Therefore, the rating range of the users in the same community as the target user belongs can be used to predict rating. For example, if the range of rating in the community of the target user is (4, 5) for item i, it can be inferred that the target user scores may be in the range of (4,5).

Suppose the predicted rating of item x that user u scored is R_{u_x} .Considering the correlation of items' rating properties information and the community properties, the predicted rating equations can be seen in Eqs. (5)-(7).

$$User_{rating}\left(x\right)=\overline{R_u}+\sum_{m\in C_u(x)}\frac{R_{mx}-\overline{R_m}}{\left|C_u\left(x\right)\right|} \tag{5}$$

$$Item_{rating}\left(x\right)=\frac{\sum\limits_{y\in Neigh(I_x)}R_y\times sim_{item}\left(x,y\right)}{\sum\limits_{y\in Neigh(I_x)}sim_{item}\left(x,y\right)} \tag{6}$$

$$R_{ux}=\left(1-\beta\right)\times User_{rating}\left(x\right)+\beta\times Item_{rating}\left(x\right) \tag{7}$$

In Eq. (5), $\overline{R_u}$ represents the average rating of the user U. C_u represents the community that the user U belongs to. $C_u(x)$ represents users in C_u who score item x. $|C_u(x)|$ represents the number of users in C_u . R_{mx} represents the rating of item x that the users $m\in C_u(x)$ scores. $\overline{R_m}$ represents the average rating. R_y represents the rating of the item y that user u scores. $Neigh(I_x)$ represents the nearest neighbor set of X. β is the control parameter.

3.2.2. Constructing the nearest neighbour set

It has been stated above that the construction of the nearest neighbour set is based on community detection. The algorithm proposed by Blondel is used for community detection, after which, each user belongs to a specific community. Suppose l communities $(C_1,C_2,...C_l)$ are obtained after community detection. The target user belongs to a specific community, but the target user may also have correlations with other communities. So the first step is to calculate the similarity between the target user and the communities. For the community $j\in[1,l]$, $\overrightarrow{C_j}$ represents the centroid vector of the j^{th} community $\overrightarrow{C_j}=\left(R_{C_j1},R_{C_j2},...R_{C_ji}\right)$. R_{C_ji} represents the rating of item i that the centroid vector of community j provides. The similarity calculation equations between target user i and the community j can be seen in Eqs. (8)-(10).

$$sim_{dsenti}\left(u_i, \mathrm{C}_j\right) = \left(1 - \sqrt{\frac{\sum_{t \in I_{u_i} \cap I_{C_j}} \left(R_{u_i t} - R_{C_j t}\right)^2}{\sum_{t \in I_{u_i} \cap I_{C_j}} R_{\max}^2}}\right) \tag{8}$$

$$sim_{corr}\left(u_i, \mathrm{C}_j\right) = \frac{\sum_{t \in I_{u_i} \cap I_{C_j}} \left(R_{u_i t} - \overline{R}_{u_i}\right)\left(R_{C_j t} - \overline{R}_{C_j}\right)}{\sqrt{\sum_{t \in I_{u_i} \cap I_{C_j}} \left(R_{u_i t} - \overline{R}_{u_i}\right)^2} \sqrt{\sum_{t \in I_{u_i} \cap I_{C_j}} \left(R_{C_j t} - \overline{R}_{C_j}\right)^2}} \tag{9}$$

$$sim\left(u_i, \mathrm{C}_j\right) = \alpha \times sim_{sdenti}\left(u_i, \mathrm{u}_j\right) + \left(1 - \alpha\right) \times sim_{corr}\left(u_i, \mathrm{C}_j\right) \tag{10}$$

In Eq. (8), (9), (10), I_{u_i} represents the items set that user u_i scores. I_{C_j} represents the items set that the users in C_j score. $R_{u_i t}$ represents the rating of item t that user u_i scores. $\overline{R}_{C_i}$ represents the average rating of u_i. $\overline{R}_{C_j}$ represents the average value of the centroid vector. The parameter α is the same as it is in Eq. (3).

It can be inferred that the community that a target user belongs to has the largest similarity with the target user. The size of the nearest neighbor set is set to K. Communities are sorted in descending according to the similarity to the target user. This method considers user-community similarity then user-user similarity until K users are chosen. Therefore, this method takes the rating information of users and the influence of the community properties into consideration.

3.2.3. Recommendation

The predicted rating equation of item x that the user u scores based on K-nearest neighbours set can be seen in the Eq. (11).

$$R_u\left(x\right) = \frac{\sum_{u' \in Neigh(u)} \left[sim\left(u, \mathrm{u}'\right) \times \left(R_{u'x} - \overline{R}_{u'}\right)\right]}{\sum_{u' \in Neigh(u)} \left|sim\left(u, \mathrm{u}'\right)\right|} \tag{11}$$

In Eq. (11), Neigh(u) represents the K-nearest neighbors set. $sim\left(u, \mathrm{u}'\right)$ represents the similarity between user u and u'. If the rating information of the item x that the user scores exists, then $R_{u'x}$ represents the rating that user u' scores on x. If the rating information of item x that the user u' scores does not exist, then $R_{u'x}$ = R_{ux}. The detailed information of R_{ux} can be seen in Eq. (6), $\overline{u}_{ux}$ representing the average rating.

4. Performance Evaluations

4.1. Data sets

MovieLens data sets provided by Grouplens group were taken for the experiments. They collected movie data sets from the MovieLens website: http://movielens.org and publish these data sets on the website: http://grouplens.org/datasets/ movielens. Ml-100k data set included 100000 ratings [1, 5] from 943 users on 1682 movies is taken for experiments. Besides that, a shell script named mku.sh is used to generate all training data sets and test data sets. Through setting parameters in mku.sh, 5 training data sets including u.base1, u.base2, u.base3, u.base4, u.base5 and 5 test data sets including u.test1, u.test2, u.test3, u.test4, and u.test5 are generated. The ratio of training data sets and test data sets is 4:1. Data crossover phenomenon does not exist between paired training data sets and test data sets.

In this paper, the new collaborative filtering algorithm based on community detection (DSRCD) is taken for experiments. The first task of community detection is to build network. U.data was used as the raw data and built the user-user network. In user-user network, nodes represent 943 users and the lines among these nodes are the similarities between users.

4.2. Evaluation Criteria

Considering the recommendation accuracy effectiveness of the algorithm, Mean Absolute Error (MAE) is taken to evaluate the performance of the algorithm. Through comparing the difference between the predicted value and the user rating scores, the formula is given in Eq. (12).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{i=1}^{t}\left|R_{ui} - p_{ui}\right|}{t} \tag{12}$$

In Eq. (9), n represents the number of users, and t represents the number of the items evaluated by a specific user. R_{ui} represents the real rating of item i that the user u scores. p_{ui} represents the predicted rating of the item i for the user u scores. Eq. (12) indicates that the closer the real ratings of the items and the predicted ratings of the items are, the smaller the value of MAE is. Therefore, MAE can be used to evaluate the accuracy of the algorithm.

4.3. Experiments

DSRCD was compared with traditional collaborative filtering algorithm and the algorithm proposed by Zhao. These three algorithms will be tested to get the value of MAE at different K-nearest neighbour candidate sets and different data density. Data density parameter σ represents the ratio between the number of training data sets and the number of the whole data sets. First, parameter α in similarity calculation equation and parameter β in predicted rating were tested to shown their influences on MAE. Then the parameter σ was tested. U.base1, u.base2, u.base3, u.base4, u.base5 are taken as training data sets, u.test1, u.test2, u.test3, u.test4, u.test5 are taken as test data sets. The designed strategies are as following.

4.3.1. The influence α of on MAE

Given K=20, the influence of α on MAE was tested. α was assigned the following six values: 0, 0.2, 0.4, 0.6, 0.8, 1. The result of MAE can be seen in Figure 1. In Table 2, when α =0.2 or α =0.4, the values of MAE were relatively small compared with other values. When α =0, the similarity calculation equation became the Pearson similarity equation. When α =0.2, the average value of MAE had the least value. This illustrated that when the similarity calculation equation took data sensitivity into consideration, the accuracy of the recommendation became higher. In similarity calculation equation, the part of Pearson similarity calculation played the major role.

Table 2. the influence of α on MAE

Test α	U.test1	U.test2	U.test3	U.test4	U.test5	Average
0	0.8122	0.8073	0.7928	0.8056	0.8174	0.8071
0.2	0.8032	0.7995	0.7911	0.8117	0.8174	0.8046
0.4	0.8149	0.8063	0.7968	0.8017	0.8087	0.8057
0.6	0.8159	0.8077	0.7972	0.8124	0.8163	0.8099
0.8	0.8167	0.8093	0.7987	0.8133	0.8166	0.8110
1	0.8174	0.8093	0.7990	0.8155	0.8166	0.8115

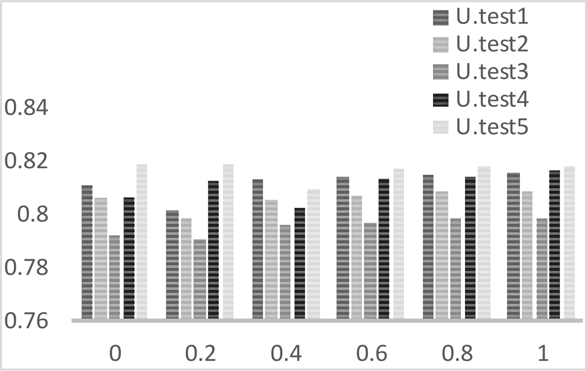

Figure 1. The influence of α on MAE

4.3.2. The influence of β on MAE

Given α = 0.2 and K = 20, the influence of β on MAE was tested. β took the following six values: 0,0.2,0.4,0.6,0.8,1. The results of MAE were calculated and shown in Table 3 and Figure 2. When β = 0, the average of MAE equals 0.7941, which had the least value. It is shown in table 3 that when β is getting larger, the values of MAE are also getting larger. This indicates that using the properties of community clustering to predict rating is better than using the properties of item rating to do the same work.

Table 3. the influence of β on MAE

Test β	U.test1	U.test2	U.test3	U.test4	U.test5	Average
0	0.7934	0.7883	0.7807	0.8032	0.8066	0.7944
0.2	0.7961	0.7892	0.7807	0.8047	0.8050	0.7951
0.4	0.7966	0.7892	0.7821	0.8055	0.8073	0.7961
0.6	0.7977	0.7911	0.7821	0.8055	0.8097	0.7972
0.8	0.7984	0.7917	0.7834	0.8056	0.8107	0.7980
1	0.7993	0.7941	0.7864	0.8079	0.8107	0.7997

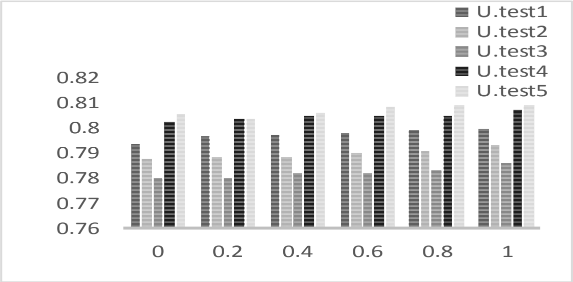

Figure 2. The influence of β on MAE

4.3.3. The influence of K and σ

The parameter K and the data density parameter σ were tested to check their influence on MAE. σ was the fraction of the number of the training sets and the sum of the number of the training sets and testing sets. In order to get the best recommendation accuracy, parameter α and parameter β were set to appropriate values. DSRCD was compared with the traditional collaborative filtering algorithm to check the difference in MAE obtained from the five test data sets. The two algorithms were tested in different K and σ, and the detailed strategies were shown as follows.

In table 4, the following data sets Ua.test, Ub.test, Uc.test, Ud.test, Ue.test were obtained according to the values of σ; the higher value of σ, the larger of the ratio between the training data sets and the whole data sets. In the above experiments, σ is set to 0.8. The results of the experiments indicate that the larger of σ, the smaller of the value of MAE, which also proves that the more training data collected, the more recommendation accuracy can be achieved.

Table 4. the influence of σ on MAE

Test	Ua.test		Ub.test		Uc.test		Ud.test		Ue.test	
σ	*N*	*T*	*N*	*T*	*N*	*T*	*N*	*T*	*N*	*T*
0.4	0.8053	0.8503	0.7962	0.8492	0.7893	0.8407	0.8096	0.8547	0.8134	0.8554
0.5	0.8011	0.8481	0.7945	0.8473	0.7866	0.8377	0.8088	0.8524	0.8112	0.8540
0.6	0.7987	0.8462	0.7922	0.8457	0.7841	0.8361	0.8073	0.8501	0.8094	0.8521
0.7	0.7945	0.8443	0.7903	0.8422	0.7824	0.8347	0.8055	0.8482	0.8086	0.8492
0.8	0.7867	0.8381	0.7853	0.8380	0.7793	0.8317	0.8004	0.8443	0.8037	0.8452

From the above experiments, it can be found that DSRCD and the traditional collaborative filtering algorithm have different K-nearest neighbours when the two algorithms achieve the best recommendation accuracy. Besides that, the values of MAE obtained by DSRCD are smaller than the values of MAE obtained by traditional collaborative filtering algorithm, no matter in which test data sets. The results of the experiments prove the effectiveness of DSRCD. When K = 30, in DSRCD the average of MAE in five test data sets equals 0.7908. Meanwhile, in the traditional collaborative filtering algorithm, when K = 30, the average of MAE in five test data sets equals 0.8385, which is larger than the average value obtained from DSRCD.

Let K=20 and data density parameter σ takes the following five values: 0.4, 0.5, 0.6, 0.7, 0.8. The result of MAE can be seen Figure 3, 4. The results of the experiments indicate that the larger of σ, the smaller the value of MAE. This also proves that the more training data collected, the more recommendation accuracy can be achieved.

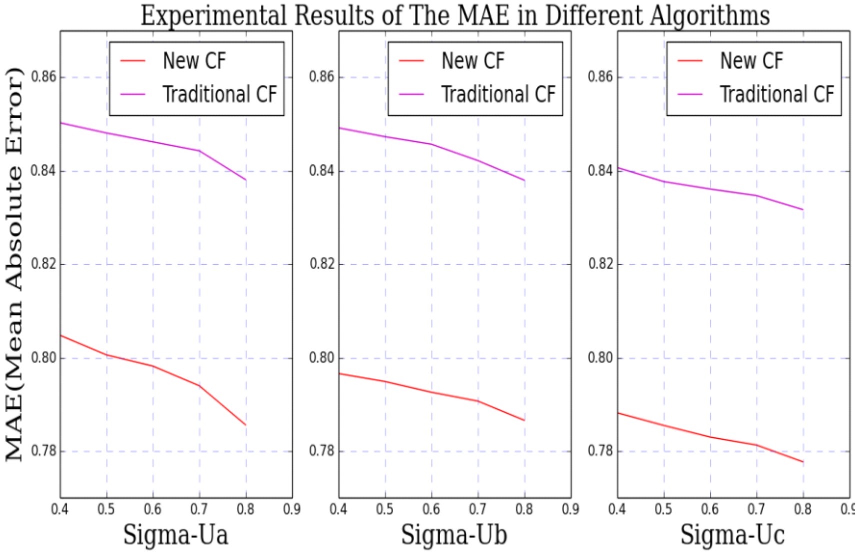

Figure 3. The influence of σ 1

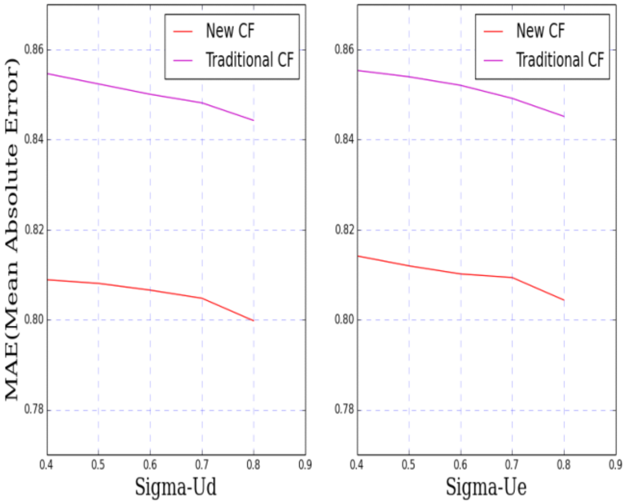

Figure 4. The influence of σ 2

Result analysis: in Figure 3, 4, the purple line represents the traditional collaborative filtering algorithm, and the red line represents DSRCD. It can be seen that the red line was always below the purple line, no matter what value σ was. The results prove that DSRCD has a better recommendation performance than traditional collaborative filtering algorithm.

4.3.4. Comparisons to Zhao's algorithm

In this subsection, DSRCD was compared with the algorithm proposed by Qinqin Zhao to check the difference in MAE. Because the algorithm proposed by Qinqin Zhao and DSRCD were both trying to amend the similarity calculation equation to get a better nearest neighbour set. The two algorithms were tested in different Ks, which took the seven values as follows: 20,25,30,35,40,45,50. Besides that, u.base1, u.base2, u.base3, u.base4, u.base5 were taken as training data sets. U.test1, u.test2, u.test3, u.test4, u.test5 were taken as test data sets. The results of MAE can be seen in Table8 and Table9.

Result analysis: in table 5, N represents the DSRCD, Z represents the algorithm proposed by Qinqin Zhao. It can be concluded that in the algorithm proposed by Qinqin Zhao, the value of MAE obtained from U.test3 achieved the least when K=35, the value of MAE obtained from the U.test4 had the least value when K=40, and the value of MAE obtained from U.test1, U.test2, U.test5 had the least value, when K=45. Through calculation, we found that when K=45, the average of MAE in 5 test data sets equals 0.7960, which was the least value. According to the above experiments, in DSRCD, when K=30, the average of MAE equals 0.7922, which was the least value.

DSRCD had better recommendation accuracy than the algorithm proposed by Zhao. When algorithms achieved the highest recommendation accuracy, the size of the nearest neighbour set may be different.

Table 5. the influence of K on MAE

Test	U.test1		U.test2		U.test3		U.test4		U.test5	
K	*N*	*Z*	*N*	*Z*	*N*	*Z*	*N*	*Z*	*N*	*Z*
20	0.7934	0.8110	0.7883	0.8021	0.7807	0.7793	0.8032	0.8037	0.8066	0.8104
25	0.7911	0.8092	0.7854	0.8003	0.7785	0.7785	0.8016	0.8021	0.8043	0.8097
30	0.7867	0.8063	0.7853	0.7987	0.7793	0.7783	0.8004	0.8015	0.8037	0.8085
35	0.7889	0.8057	0.7872	0.7978	0.7795	0.7772	0.7989	0.7997	0.8049	0.8079
40	0.7903	0.8054	0.7880	0.7943	0.7817	0.7784	0.7993	0.7970	0.8051	0.8073
45	0.7928	0.8049	0.7880	0.7931	0.7821	0.7791	0.8011	0.7983	0.8072	0.8045
50	0.7943	0.8063	0.7891	0.7983	0.7822	0.7816	0.8045	0.8012	0.8074	0.8057

5. Conclusions

DSRCD proposes a method how to use a user-item network to construct a user-user network and how to design a similarity calculation equation considering data sensitivity. In DSRCD, the community detection method is used to find the nearest neighbours candidate set in collaborative filtering, which begins to solve the cold start problem. At the same time, a forecasting method based on community detection and the attributes of the products scores is used to make the final predicting score more accurate.

DSRCD was tested to check the influences of parameter α and parameter β on MAE. DSRCD, the traditional collaborative filtering and the algorithm proposed by Zhao were compared in different K and σ. The result of the experiments indicates that DSRCD has better performance than traditional collaborative filtering and the algorithm proposed by Zhao in MAE. In the real world application, we should also use some data sets as training sets and first find the influences of parameter α and parameter β on MAE. Then find the best values for K and σ. DSRCD is complicated to implement. In the future, we should improve its efficiency. How to solve the scalability for CF is our future research problem.

Acknowledgements

This work was supported by the National Nature Science Foundation of China under contract number 61271259, the Chongqing Nature Science Foundation under contract number CTSC2011jjA40006, CTSC2012jjA40038, and the Research Project of Chongqing Education Commission under contract number KJ120501C.

The authors would like to show our deepest gratitude to Professor John E. Hopcroft, who has provided us with valuable guidance.

References

[1] Adomavicius G., Tuzhilin A., Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *Knowledge and Data Engineering, IEEE Transactions on*, **17**, 6, 2005, 734-749.

[2] Bellogín A., Cantador I., Díez F., An empirical comparison of social, collaborative filtering, and hybrid recommenders, *ACM Transactions on Intelligent Systems and Technology (TIST)*, **4**, 1, 2013, 14.

[3] Biancalana C., Gasparetti F., Micarelli A., An approach to social recommendation for context-aware mobile services, *ACM Transactions on Intelligent Systems and Technology (TIST)*, **4**, 1, 2013, 10.

[4] Blondel V D., Guillaume J L., Lambiotte R., Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **10,** 2008, 10008.

[5] Breese J.S., Heckerman D., Kadie C., Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence Morgan Kaufmann Publishers Inc*., 1998: 43-52.

[6] Balabanović M., Shoham Y., Fab: content-based, collaborative recommendation, *Communications of the Association of Computing Machinery*, **40**, 3, 1997, 66-72.

[7] Chen X.Y., Zhang C., Lin Z.Q., Xiao B., Ma H, A Collaborative Filtering Method using Topological-Potential Based Community Discovery Strategy, *Institute of Electrical and Electronics Engineers, Conference on Information Security and Artificial Intelligence*, 2010. 229-223.

[8] Guo L., Ma J., Chen Z., Learning to recommend with social relation ensemble. *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, 2599-2602.

[9] Girardi R., Marinho L.B., A domain model of Web recommender systems based on usage mining and collaborative filtering, *Requirements Engineering*, **12**, 1, 2007, 23 - 40.

[10] Getoor L., Sahami M., Using probabilistic relational models for collaborative filtering, *Workshop on Web Usage Analysis and User Profiling*, 1999.

[11] Good N., Schafer J.B., Konstan J.A., Combining collaborative filtering with personal agents for better recommendations, *Innovative Applications of Artificial Intelligence Conferences*, 1999, 439-446.

[12] Melville P., Mooney R.J., Nagarajan R., Content-boosted collaborative filtering for improved recommendations, *American Association for Artificial Intelligence*, 2002, 187-192.

[13] Jiang M., Cui P., Liu R., Social contextual recommendation, *Proceedings of the 21st ACM international conference on Information and knowledge management.* ACM, 2012, 45-54.

[14] Newman M.E.J., Fast algorithm for detecting community structure in networks, *in: Physical review E*, **69**, 6, 2004, 066133.

[15] Pavlov D., Pennock D M., A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains, *Neural Information Processing Systems Foundation*, 2002, 2, 1441-1448.

[16] Sarwar B.M., Konstan J.A., Borchers A., Herlocker J., Miller B., Riedl J., Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collabor*ative Filtering System, Proceedings of Computer Supported Cooperative Work'98, (Seattle, WA, USA),* Nov. 1998, 345-354.

[17] Shih Y.Y. and Liu D.R., Hybrid recommendation approaches: collaborative filtering via valuable content information. *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on. IEEE*, 2005, 217b-217b.

[18] Sun G.F., Wu L., Liu Q, Zhu C., Chen E.H., Recommendations Based on Collaborative Filtering by Exploiting Sequential Behaviours, *Ruan Jian Xue Bao/Journal of Software*, **24**,11, 2013, 2721-2733.

[19] Su X.P., Song Y.R., Lou J.G., Jiang Y.L., Leveraging Overlapping Communities Detection Improve Personalized Recommendation in Folksonomy Networks, *Journal of Chinese Computer Systems*, **34**, 9, 2013, 2036-2041.

[20] Tang J., Zhang Y., Sun J.M., Rao J.H., Yu W.J., Chen Y.R., and Fong A.C.M., Quantitative Study of Individual Emotional States in Social Networks, T, *Affective Computing*, **3**, 2, 2012, 132-144.

[21] Yang D.Q., Zhang D.Q., Yu Z.Y., Yu Z.W., Fine-grained preference aware location search leveraging crowd sourced digital footprints from LBSNs, *in 13th International Conference on Ubiquitous Computing'13, (Zurich, Switzerland*), Sept. 2013, 479-488.

[22] Yoshii K., Goto M., Komatani K., An efficient hybrid music recommender system using an increment ally trainable probabilistic generative model, *IEEE Transactions on Audio Speech and Language Processing*, **16**, 2, 2008, 435-447

[23] Zhu L., Ge W., Research on Personalized Recommendation Algorithm Based on Social Network, *International Conference on Computer and Electrical Engineering 4th ASME Press*, 2011.

[24] Zhao Q.Q., K. Lu, Wang B., SPCF: A Memory Based Collaborative Filtering Algortihm via Propagation, *Chinese Journal of Computers*, **36**, 3, 2013, 671–676.

[25] Zhang Y., Zhang B., Gao K.N., Guo P.W., Sun D.M., Autonomy Oriented Personalized Tag Recommendation, *Journal of Electronic*, **40**, 12, 2012, 2353-2359.

Received 21.10.2014, accepted 6.03.2015