Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

The New Similarity Measure Based on User Preference Models for Collaborative Filtering

Qiao Cheng, Xiangke Wang, Dong Yin, Yifeng Niu, Xiaojia Xiang, Jian Yang, Lincheng Shen
College of Mechatronics and Automation
National University of Defense Technology
Changsha, Hunan Province, China
qiaochengsci@sina.com

***Abstract*— The recommender system is widely used in many areas in the age of information overload. Collaborative filtering (CF), as one of the most successful methods used for recommendation, recommends items based on the nearest neighbors of the target user. Thus, the performance of the recommender system depends largely on the similarity measure used for selecting neighbors. Most of the traditional similarity measures are based on the rating data that users give to the items, such as Pearson's correlation and cosine, and suffer from low performances. In order to improve the performance of CF recommender system, this paper proposes a new kind of similarity measure based on user preference models and applies it to a movie recommender system. In this paper, two user preference models are build, one is focused on percentages of different movie genres a user has watched, the other is on the average ratings a user has given to different genres of movies. Based on the two user preference models, two new similarity measures are designed. The experiments compare the performance of the two new similarity measures with the Pearson's correlation which is widely used in the traditional CF, and are carried out on the MovieLens data set. The results show that the new similarity measure based on the watched genre ratio model outperforms that of traditional CF in all aspects. While the new similarity measure based on the average genre rating model has almost the same performance with the traditional one, but has much less computing complexity for no need in finding the co-rated items.**

***Index Terms*— Similarity measure, user preference model, collaborative filtering, recommender system.**

I. Introduction

Nowadays, we are living in a age of information overload. Even with the help of search engine, we can get inundated in the information flood. Recommender system is a good tool that can relieve such kind of dilemma by providing users with personalized information. We can see many kinds of such applications in areas like e-commerce, movie website, music website, travel agency and so on.

As the recommender system has played an increasing role in our life, numerous method that improve the performance of recommender system have been developed recent years. Collaborative Filtering (CF) is one of the most successful methods used in recommendation. In CF recommender system, those users who have the most similar preferences with the target user are called the nearest neighbors of the target user. The main idea of the CF is to recommend the target user with items that the nearest neighbors have accessed but the target user has not, for they will share many items due to similar preferences. Thus, the most critical thing in CF approach is to find the nearest neighbors that match the preference of the target user most. That's to say, the similarity measure in CF is very critical to its performance. In traditional CF recommender system, there are many similarity measures already, such as Pearson's correlation, consin and other similarity measures originated from these similarity measures. Even so, there is still some room for the improvement of recommender system performance, as well as the similarity measures accuracy in selecting nearest neighbors.

The objective of our study is to propose new similarity measure to improve the accuracy of the recommendation. Unlike the traditional similarity measures that rely on the rating data very much, our new similarity measures focus more on the item types which can express the knowledge of user preference more accurately. Our approach is validated by experiments on MovieLens data set for the effectiveness.

The rest of this paper is organized as follows. Section 2 gives a brief overview of related works on the similarity measures used in collaborative filtering. In section 3, we first address the weakness of traditional similarity measures and our idea to improve those weakness. Next, two user preference models are build in detail, as well as two similarity measures and prediction methods based on the two user preference models, respectively. In section 4, the experiments comparing the performance of our proposed similarity measures with the traditional CF similarity measure called Pearson's correlation are conducted, followed by the discussion on the results. In section 5, a summary and the further research issues are presented.

II. Literature review

Since the quality of similarity measure has a large impact on the recommendation accuracy, many improvements on the similarity measure have been made. one of those improvement is the use of a *k*-nearest neighbor (*k*-NN) graph. Lien et al. [1] propose a new similarity measure based on graph models, which calculated the similarity from connections on a graph with vertices being users and products. In order to reduce the time in finding top-*k* similar neighbors, Park et al. [2] present a rapid CF algorithm called Reversed CF (RCF) which utilizes a *k*-nearest neighbor graph. This approach reverses the process of finding *k* neighbors, which finds the *k*-nearest neighbors of rated items instead of finding *k* similar neighbors of unrated items.

Another improvement is based on the use of fuzzy set technique. Wang et al. [3] use fuzzy set theory to express the contribution of rating difference and propose an innovative fuzzy similarity measure (FSM). The FSM is the weighted average by all valid fuzzy rating similarity (FRS). Combined with a user-relevant aggregation (URA) for prediction, FSM-URA approach significantly improves the prediction accuracy comparing to the existing recommendation approaches as shown in the experiments. AlShamri et al. [4] use fuzzy logic to weight some variables of the user profile, then apply these weights to cosine similarity measure and Pearson correlation coefficient and uses the fuzzy-weighted cosine similarity measure and fuzzy-weighted Pearson correlation coefficient for the neighbor selecting.

There are some other methods in the improvement of similarity measure. Ahn [5] proposes a heuristic measure called PIP which is composed of three factors of similarity, Proximity, Impact and Population. Based on Ahn's work, Liu et al. [6] analyze the drawbacks of PIP and proposed an improved PIP measure as PSS (Proximity-Significance-Singularity). Apart from these, Hu [7] proposes a novel similarity-based perception using nonadditive indifference indices to estimate an overall rating that a user would give to a specific item. When exploring coefficients that preform better than Pearson correlation coefficient, AlShamri [8] proposes a more general coefficient called Power coefficient based on Jaccard and Dice coefficients. The power coefficient takes into consideration that users have both positive and negative matches and therefore those coefficients like Jaccard and Dice coefficients have to be modified to take negative matches into consideration.

Choi et al. [9] consider it more reasonable that the similarity between a target item and each of the co-rated items should be considered when finding neighbors of a target user, and a different set of neighbors should be selected for each different target item. Therefore, they propose a new similarity function in which the rating of a user on an item is weighted by the item similarity between the item and the target item. Rouzbeh et al. [10] use Linked Open Data (LOD) for semantic analysis of items and propose a similarity measure that combines the semantic analysis of items with collaborative filtering approaches. The new similarity measure is derived by computing the partitioned information content (PIC) of shared and distinctive features of two resources. Deng et al. [11] design a Item-attribute Based Hybrid Filtering (IBHF) based on movie features of the multimedia information. In the IBHF, the similarity measure is improved based on Pearson's correlation and integrates the influence brought by difference in movie attributes like the category, publishing time and other properties. This similarity measure takes consideration of the movie attributes and performances slightly better than the classic item-based collaborative filtering, but the calculation of the similarity can be much complex and time consuming, and the accuracy depends much on the adjusting of the parameters, which can bring complexity to the recommendation process.

III. The new similarity measure

In this section, we first introduce the motivation and assumptions of proposed similarity measure approaches. Then, we build our user preference models. Finally, we present the new similarity measures and the prediction model for the recommendation.

A. The motivation

In most of the traditional recommender systems, the similarities between users are calculated based on the Pearson correlation coefficient or cosine similarity, which are defined by (1) and (2), respectively.

$$Sim(u,n)_P = \frac{\sum_{i\in C_{u,n}}(r_{ui}-\overline{r_u})(r_{ni}-\overline{r_n})}{\sqrt{\sum_{i\in C_{u,n}}(r_{ui}-\overline{r_u})^2}\sqrt{\sum_{i\in C_{u,n}}(r_{ni}-\overline{r_n})^2}} \tag{1}$$

$$Sim(u,n)_C = \frac{\sum_{i\in C_{u,n}}(r_{ui})(r_{ni})}{\sqrt{\sum_{i\in C_{u,n}}(r_{ui})^2}\sqrt{\sum_{i\in C_{u,n}}(r_{ni})^2}} \tag{2}$$

The u and n denotes two users, r_{ui} is the rating that user u gives to item i, $\overline{r_u}$ denotes the average rating that user u gives to the items, the symbol $C_{u,n}$ denotes the set of items that user u and n have co-rated.

From our literature review section, we can know that most of the improvements are based on these traditional similarity measures. However, those similarity measures have some drawbacks or limitations.

Firstly, almost all the similarity measures rely on the rating data. However, to rate for every movie that a user have watched is very time-consuming. Thus, not every user is willing to do such a thing. Besides, not everyone can rate carefully, that's to say, some people will give a untrue rating which will influence the similarity calculation.

Secondly, some similarity measures are too complex and need too much calculation.

Thirdly, the user preference models are expressed by the rating data, which is too implicit and can not convey the preference information directly.

Finally, there are still some room for improvement in the accuracy of the recommendation.

Therefore, we propose the new similarity measures that rely little on the rating data and the calculation is not too complex. In order to express user's preference more directly, we build two user preference models that are very easy for understand. Finally, we design the similarity measures that aim at improving the performance of the recommender system.

B. User preference models

In common collaborative filtering recommender system, user preference is usually expressed by the rating data. In virtue of such kind of preference, we can know how each movie watched by the user fit the user's appetite, but have no idea of other movies. On the contrary, our approaches care more about what kind of movies that fit a user's appetite and how all kind of movies fit a user's appetite. We use the MovieLens data set for the analysis. There are 19 genres for the movies in this data set including the unknown genre. In the long run, a user usually has a fix appetite for some specific movie genres. Thus, we use a vector to express a user's preference and each element of the vector represents a specific feature of a movie genre. Due to the different feature that the elements of preference vector could represent, we build two user preference models, one is to express the proportion of each genre that a user have watched, and the other one is to represent the average rating for each genre from the user. Here we give a detail description about how these two preference models are build.

1) Preference model based on watched genre ratio: Generally, a user watches the most preferred genre of movies most. Therefore, we aim at getting the knowledge about which genres a user watched most in our first user preference model.

we use vector $\overrightarrow{G^{(i)}} = (g_0^{(i)}, g_1^{(i)}, ..., g_k^{(i)}, ..., g_{M-1}^{(i)}), (k = 0, 1, ..., M-1)$ to represent the genre information of movie i. M is the number of genres. Each element of $\overrightarrow{G^{(i)}}$, $g_k^{(i)}$, denotes whether movie *i* can be classified into the k*th* movie genre. The value of $g_k^{(i)}$ is set to be 1 if the movie *i* can be classified into the k*th* movie genre and 0 otherwise, as described in (3). Table I displays the exact meaning of each element $g_k^{(i)} (i = 0, 1, ..., M-1)$ in $\overrightarrow{G^{(i)}}$. Thus, every item (movie) *i* can be expressed by a vector $\overrightarrow{G^{(i)}}$.

$$g_k^{(i)} = \begin{cases} 1, i \in G_k \\ 0, otherwise \end{cases} \tag{3}$$

where G_k represents the set of movies that belong to the k*th* genre.

For user *u*, we use the symbol $W^{(u)}$ to denote the set of movies that user *u* has watched and $\overrightarrow{Pa^{(u)}} = (pa_0^{(u)}, pa_1^{(u)}, ..., pa_k^{(u)}, ..., pa_{M-1}^{(u)}), (k = 0, 1, ..., M-1)$ to

TABLE I

MEANING OF $g_k^{(i)}$

$g_k^{(i)}$	$g_0^{(i)}$	$g_1^{(i)}$	$g_2^{(i)}$	$g_3^{(i)}$	$g_4^{(i)}$
Genre	unknown	Action	Adventure	Animation	Children's
$g_k^{(i)}$	$g_5^{(i)}$	$g_6^{(i)}$	$g_7^{(i)}$	$g_8^{(i)}$	$g_9^{(i)}$
Genre	Comedy	Crime	Documentary	Drama	Fantasy
$g_k^{(i)}$	$g_{10}^{(i)}$	$g_{11}^{(i)}$	$g_{12}^{(i)}$	$g_{13}^{(i)}$	$g_{14}^{(i)}$
Genre	Film−Noir	Horror	Musical	Mystery	Romance
$g_k^{(i)}$	$g_{15}^{(i)}$	$g_{16}^{(i)}$	$g_{17}^{(i)}$	$g_{18}^{(i)}$	
Genre	Sci−Fi	Thriller	War	Western	

denote the first type of preference model of user *u*. From the set $W^{(u)}$, we can calculate the total movie number of each genre that user *u* has watched. Let $\overrightarrow{N_g^{(u)}}$ denote this variable. Thus, $\overrightarrow{N_g^{(u)}}$ can be calculated by (4). $W_k^{(u)} = \{i | i \in W^{(u)}, and\ g_k^{(i)} = 1\}$, is the set of movies that has been watch by user *u* and can be classified into the k*th* genre. $pa_k^{(u)}$ is calculated by (6). Obviously, $pa_k^{(u)}$ represents the percentage of the k*th* genre that user *u* has watched. After calculated all the $pa_k^{(u)}, (k = 0, 1, ..., M-1)$, we can get our first user preference model $\overrightarrow{Pa^{(u)}}$.

$$\begin{aligned} \overrightarrow{N_g^{(u)}} &= \sum_{i \in W^{(u)}} \overrightarrow{G^{(i)}} \\ &= (|W_0^{(u)}|, |W_1^{(u)}|, ..., |W_k^{(u)}|, ..., |W_{M-1}^{(u)}|) \end{aligned} \tag{4}$$

$$|W_k^{(u)}| = \sum_{i \in W^{(u)}} g_k^{(i)} \tag{5}$$

$$pa_k^{(u)} = \frac{|W_k^{(u)}|}{|W^{(u)}|} \tag{6}$$

where $|W|$ denotes the number of elements in set W.

2) Preference model based on genre average rating: The main idea of the second preference model is that the average rating to a certain movie genre can convey some information about how much the user likes this genre of movies. We use $\overrightarrow{Pb^{(u)}} = (pb_0^{(u)}, pb_1^{(u)}, ..., pb_k^{(u)}, ..., pb_{M-1}^{(u)}), (k = 0, 1, ..., M-1)$ denote the second type of preference model of user *u*, and $r_{u,i}$ denotes the rating given to movie *i* by user *u*. Thus, $pb_k^{(u)}$ can be calculated by (7).

$$pb_k^{(u)} = \frac{\sum_{i \in W_k^{(u)}} r_{u,i}}{|W_k^{(u)}|} \tag{7}$$

Until now, we have build two user preference models $\overrightarrow{Pa^{(u)}}$ and $\overrightarrow{Pb^{(u)}}$. Next, we will use these two model to define our similarity measures.

C. Similarity measures and recommendation

The similarity is to measure how much similar the preferences of two users are. As in previous subsection, we have build two kind of user preference models and both of the two models are expressed by vectors. Therefor, we define the similarity measures as the distance between the preference vectors of two users. Based on model $\overrightarrow{Pa^{(u)}}$ and $\overrightarrow{Pb^{(u)}}$, we define the similarity between user *u* and user *n* as (8) and (9), respectively.

$$Sim(u,n)_a = \sqrt{\sum_{k=0}^{M-1} (pa_k^{(u)} - pa_k^{(n)})^2} \tag{8}$$

$$Sim(u,n)_b = \sqrt{\sum_{k=0}^{M-1} (pb_k^{(u)} - pb_k^{(n)})^2} \tag{9}$$

Since we have defined two new similarity measures, we can find out the nearest neighbors of a target user, say user *u*. As collaborative filtering, we use (8) or (9) to calculate the similarities between user *u* and all other users, then the *k* users with the highest similarities are the top-*k* neighbors of user *u*. The set of *k*-nearest neighbors that selected by similarity $Sim(u,n)_a$ is denoted by $neighbors(u)_a$, and those selected by similarity $Sim(u,n)_b$ is denoted by $neighbors(u)_b$. After that, we can predict the potential movies that may interest the target user. Let $Candidate_a$ and $Candidate_b$ denote the sets of movies that the target user has not watched but the *k*-nearest neighbors have. The difference between $Candidate_a$ and $Candidate_b$ is that they are obtained from the *k*-nearest neighborhood with similarity $Sim(u,n)_a$ and $Sim(u,n)_b$, respectively. That's to say, $Candidate_a = \bigcup_{n \in neighbors(u)_a} W^{(n)} - W^{(u)}$ and $Candidate_b = \bigcup_{n \in neighbors(u)_b} W^{(n)} - W^{(u)}$. Based on the two *k*-nearest neighborhood sets $neighbors(u)_a$ and $neighbors(u)_b$, we define the prediction function by (10) and (11), respectively.

$$\begin{cases} pred(u,i)_a = \overline{r_u} + \frac{\sum_{n \in neighbors(u)_a} Sim(u,n)_a \bullet (r_{n,i} - \overline{r_n})}{\sum_{n \in neighbors(u)_a} Sim(u,n)_a} \\ \forall i \in Candidate_a \end{cases} \tag{10}$$

$$\begin{cases} pred(u,i)_b = \overline{r_u} + \frac{\sum_{n \in neighbors(u)_b} Sim(u,n)_b \bullet (r_{n,i} - \overline{r_n})}{\sum_{n \in neighbors(u)_b} Sim(u,n)_a} \\ \forall i \in Candidate_b \end{cases} \tag{11}$$

By the prediction function (10) or (11), we can recommend the top N movies with the highest prediction ratings to the target user.

IV. EXPERIMENTS

In order to prove the performance of our two new similarity measures, several experiments were performed. We use the MovieLens data set for comparisons. There are 100,000 ratings (range from 1 to 5) from 943 users on 1682 movies, each user has rated at least 20 movies.

A. Evaluation metrics

Precision and recall are two metrics that are commonly used to measure the quality of a recommendation. From the aspect of the recommendation results, items (movies) can be classified into items that recommended and unrecommended. While from the aspect of the results in the testset, movies can be classified into movies that the target user watched and unwatched. Table II displays these classifications and the relationship between them. Let $R(u)$ denote the set of movies that recommended to user *u*, and $T(u)$ denote the set of movies that the user *u* watched in the testset. The Precision and Recall are defined by (12) and (13), respectively.

TABLE II

ITEMS CLASSIFICATION FOR EVALUATION

		Recommendation Results	
		Recommended	Unrecommended
Results in testset	watched	True Positive (TP)	False Negative (FN)
	unwatched	False Positive (FP)	True Negative (TN)

$$Precision = \frac{|TP|}{|TP| + |FP|} = \frac{|R(u) \bigcap T(u)|}{|R(u)|} \tag{12}$$

$$Recall = \frac{|TP|}{|TP| + |FN|} = \frac{|R(u) \bigcap T(u)|}{|T(u)|} \tag{13}$$

Precision is the proportion of correctly recommended items in the whole recommendation list of the target user, while recall is the proportion of correctly recommended items in the whole watched list of the target user. As the number of recommended items is a variable set by us, it seems that increasing this variable N will reduce the precision and increase the recall. Therefore, we also use another metric F1 for the evaluation. F1-metric could balance the trade-off between precision and recall, which is defined by (14).

$$F1 = \frac{2 \times recall \times precision}{recall + precision} \tag{14}$$

B. performance comparison

In the input parameters of recommender systems, *k* denotes the number of nearest neighbors chosen for recommendation, and N denotes the number of recommended items in each recommendation. These two parameter *k* and N can influence the performance of the recommender systems. We use the Pearson's correlation as our comparison method, which is the most traditional collaborative filtering method and I have described it in [12] in detail and carried it out on another dataset. We use five fold validation to get the evaluation results. Each pair of training set and testing set is $80\%/20\%$ splits of the MovieLens data set, and the splitting are randomly. For every pair of data set, each metric is computed for each user, and the average value computed for each pair of data set. Thus, there are five results for each metric. Due to the limited space, we calculated the average value of the five values for each metric.

In our first set of experiments, we set $N = 20$, and change the k from 5 to 30. The precision metric is shown by Fig. 1, and the recall metric by Fig. 2, and the F1-metric by Fig. 3. The "CF" denotes the results from the traditional collaborative filtering by Pearson's correlation, the "S1" denotes the results from our first kind of similarity measure $Sim(u,n)_a$, and the "S2" denotes our second one $Sim(u,n)_b$.

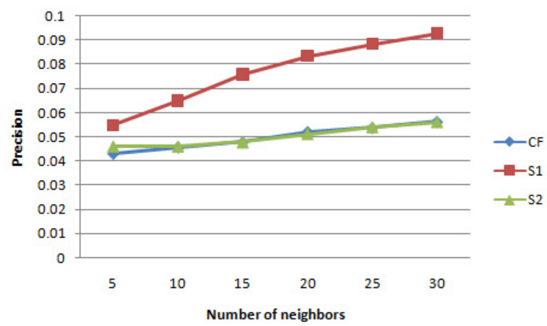

Fig. 1. Comparison of Precision with respect to k

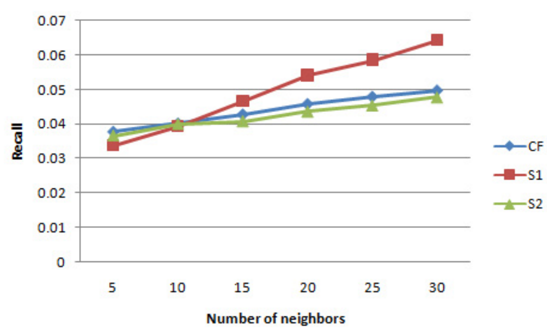

Fig. 2. Comparison of Recall with respect to k

From Fig. 1 to Fig. 3, we can know that these three metrics increases as the number of neighbors increased for all the three kind of similarity measures. The precision and F1 of S1 are higher than that of CF in all cases, while all the three metrics for S2 are almost the same with those of CF. The recall of S1 is lower than that of CF when the number of neighbors is less than 10, but increases faster after that.

In our second set of experiments, we set $k = 15$, and change the N from 10 to 35. The precision, recall and F1 metric are shown by Fig. 4, Fig. 5 and Fig. 6, respectively. From these figures, we can see that the precisions of all the three methods decreases as the number of recommendation increased, but the precision of S1 is much higher than that of CF in all cases. The recall seems to be almost linear relevant with the number of recommendation, all the recalls of the three methods are growing as the increase in the number of recommendation. Still, the F1 of S1 is much higher than that of CF. The F1 of S2 is slightly lower than that of CF, but surpass that of CF when the number of recommendation become bigger than 30.

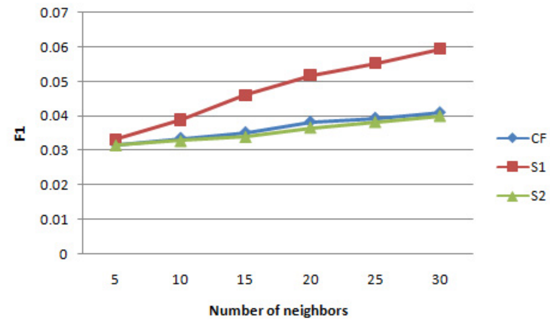

Fig. 3. Comparison of F1 with respect to k

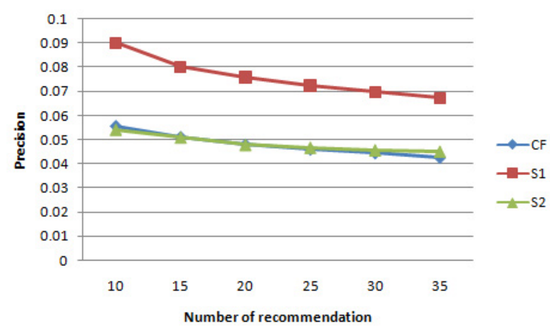

Fig. 4. Comparison of Precision with respect to N

V. Conclusions

We have proposed two new kind of similarity measures which are based on the user preference models rather than on the information of co-rating items. In this paper, we have build two user preference models. The first model is about the ratio of each movie genre in all movies watched by the target user, while the second model is about the average rating of each movie genre watched by the target user. Both the user preference models are expressed by the form of vectors. The new similarity measures are the distance of these vectors. We have designed experiments to validate the performance

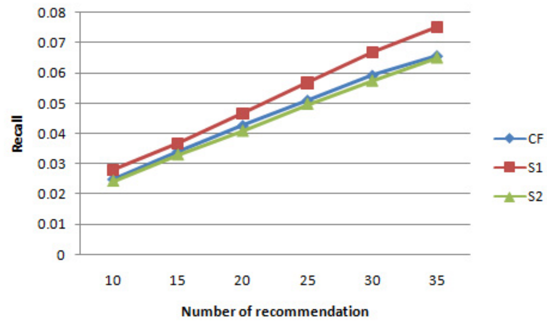

Fig. 5. Comparison of Recall with respect to N

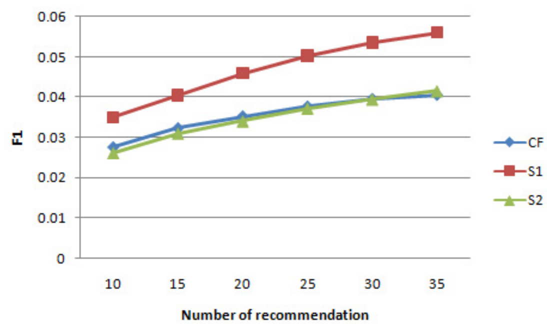

Fig. 6. Comparison of F1 with respect to N

of the proposed similarity measures. The similarity measure based on the former user preference model has much better performance than that of the traditional collaborative filtering recommender system based on Pearson's correlation. However, the similarity measure based on the latter user preference model has almost the same performance with the traditional one, but can become better when the number of neighbors or the number of recommendation is large enough. Even though, the similarity based on the second user preference model has much less computation complexity than the traditional one, which do not need to find the co-rated items when calculating the similarities as the traditional collaborative filtering do. However, our prediction function still very similar with the traditional one. It may be better to have some improvement on this aspect. This is left for the future work.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Editor and anonymous reviewers for their valuable comments and constructive suggestions. This work was supported in part by the Natural Science Foundation of China (61403406 and 61403410).

REFERENCES

[1] D. T. Lien, N. D. Phuong, "Collabotative filtering with a graph-based similarity measure," International Conference on Computing, Management and Telecommunications, pp.251-256, 2014.

[2] Y. K. Park, S. C. Park, W. S. Jung, S. G. Lee, "Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph," Expert Systems with Applications, vol. 42, no.8 pp.4022-4028, 2015.

[3] W. Wang, J. Lu, G. Q. Zhang, "A New Similarity Measure-Based Collaborative Filtering Approach for Recommender Systems," Foundations of Intelligent Systems, Springer Berlin Heidelberg. vol.277, pp.443-452, 2014.

[4] M. Y. H. Al-Shamri and N. H. Al-Ashwal,"Fuzzy-Weighted Similarity Measures for Memory-Based Collaborative Recommender Systems," Journal of Intelligent Learning Systems and Applications, vol.6, no.1, pp.1-10, 2014.

[5] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," Information Sciences, vol.178, no.1, pp.37-51, 2008.

[6] H. F. Liu, Z. Hu, A. Mian, H. Tian, X. Z. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Knowledge-Based Systems,vol.56, no.0, pp.156-166, 2014.

[7] Y. C. Hu, "Nonadditive similarity-based single-layer perceptron for multi-criteria collaborative filtering," Neurocomputing, vol.129, no.0, pp.306-314, 2014.

[8] M. Y. H. Al-Shamri, "Power coefficient as a similarity measure for memory-based collaborative recommender systems," Expert Systems with Applications, vol.41, no.13, pp.5680-5688, 2014.

[9] K. Choi,Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," Knowledge-Based Systems, vol.37, no.0, pp.146-153, 2012.

[10] M. Rouzbeh, G. D. Joseph, "Enhancing Recommender Systems Using Linked Open Data-Based Semantic Analysis of Items," Proceedings of the 3rd Australasian Web Conference (AWC 2015), Sydney, Australia, pp.11-17, 2015.

[11] Z. R. Deng, X. Zhang, X. Deng, L. Xu, and W. M. Huang, "An improvement of video recommender similarity measurement model," International Conference on Automation, Mechanical Control and Computational Engineering(AMCCE 2015), pp.675-680, 2015.

[12] Q. Cheng, J. Huang, J. X. Gong, J. G. Hao, "Simulation resource recommendation system based on collaborative filtering." 2013 Chinese Automation Congress (CAC 2013), pp.448-452, 2013.