Knowledge-Based Systems 37 (2013) 146–153

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

A new similarity function for selecting neighbors for each target item in collaborative filtering

Keunho Choi, Yongmoo Suh *

Business School, Korea University, Anam-Ro 145, Seongbuk-Gu, Seoul 136-701, Republic of Korea

ARTICLE INFO

Article history:
Received 28 November 2011
Received in revised form 25 July 2012
Accepted 25 July 2012
Available online 8 August 2012

Keywords:
Recommendation system
Collaborative filtering
Item similarity
Information overload

ABSTRACT

As one of the collaborative filtering (CF) techniques, memory-based CF technique which recommends items to users based on rating information of like-minded users (called neighbors) has been widely used and has also proven to be useful in many practices in the age of information overload. However, there is still considerable room for improving the quality of recommendation. Shortly, similarity functions in traditional CF compute a similarity between a target user and the other user without considering a target item. More specifically, they give an equal weight to each of the co-rated items rated by both users. Neighbors of a target user, therefore, are identical for all target items. However, a reasonable assumption is that the similarity between a target item and each of the co-rated items should be considered when finding neighbors of a target user. Additionally, a different set of neighbors should be selected for each different target item. Thus, the objective of this paper is to propose a new similarity function in order to select different neighbors for each different target item. In the new similarity function, the rating of a user on an item is weighted by the item similarity between the item and the target item. Experimental results from MovieLens dataset and Netflix dataset provide evidence that our recommender model considerably outperforms the traditional CF-based recommender model.

1. Introduction

In the age of information overload, the importance of personalized recommendation systems is growing. The reason for this increase is because such systems enable buyers to find what they want immediately without wasting their time and also allow sellers to provide buyers with the items they are likely to purchase. As an effort to deal with this growing significance, several techniques for developing the recommendation systems have been studied. They include content-based filtering (CBF), collaborative filtering (CF), and association rule or sequential pattern analysis. Among them, the CF technique has been widely used due to its simplicity and effectiveness, and has also proven to be useful in many practices.

In traditional CF, items are recommended based on rating information of like-minded users (known as neighbors) on the items. Therefore, the ability to identify neighbors is the most significant part of CF for improving recommendation quality [7,8]. Most CF-based recommendation systems typically identify neighbors, who rate items similarly to a target user by using similarity functions such as the Pearson correlation coefficient, cosine similarity, or distance-based similarity. Such similarity functions represent the similarity between a target user and the other user based on the rating information of items rated by both users. In this process, traditional CF gives an equal weight to each of the co-rated items when calculating the similarity between two users without considering the similarity between a target item and each co-rated item. Thus, neighbors of a target user are identical for all target items. However, a reasonable assumption is that a target user's neighbors should be different for a different target item. When the target item is a horror movie, for example, a user who has a similar preference on horror movie needs to be considered as a more important neighbor than the other user who has a similar preference on movies of other genres. Therefore, it is reasonable that item similarity between a target item and each of the co-rated items should be considered when finding neighbors of a target user. Moreover, a different set of neighbors should be selected for each different target item.

Although a number of studies have enhanced the effectiveness of CF-based recommendation systems by addressing issues, such as the new user problem (also referred to as a cold start problem) [18,32], the new item problem (also known as the first rater problem) [4,26], and the sparsity problem [15,18,25,32], few studies have addressed the issue mentioned in this paper.

The objective of our study is to propose a new similarity function in order to select different neighbors for each different target item. In the new similarity function, the rating of a user on an item

* Corresponding author. Tel.: +82 2 3290 1945; fax: +82 2 922 7220.
E-mail addresses: keunho@korea.ac.kr (K. Choi), ymsuh@korea.ac.kr (Y. Suh).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2012.07.019

is weighted by the item similarity between the item and the target item. The similarity function is expected to result in better recommendation. To that end, we first calculated item similarity between a target item and each of the co-rated items and then used it as the weight when calculating user similarity between a target user and other user. Experiments for validating the effectiveness of our approach were conducted using both *MovieLens* dataset and *Netflix* dataset.

The rest of this paper is organized as follows. Section 2 reviews previous studies regarding recommendation systems. Section 3 addresses the weakness of traditional collaborative filtering and an approach to realizing our idea. Section 4 explains how our approach makes a difference based on the results from experiments and also describes the implications of the experiments. The last section concludes our paper with summary, implications, and limitation.

2. Literature review

To date, many recommendation systems have been developed and most of them have used content-based filtering (CBF) and/or collaborative filtering (CF) as their fundamental techniques.

CBF recommendation systems typically (1) build an item profile by extracting a set of features from each item in the item set, (2) build a content-based user profile from a set of features of the items which each user purchased, (3) calculate the similarity between the user profiles and the item profiles using similarity function and (4) recommend top *n* items with high similarity scores. Specifically, they recommend items based on the similarity between items. In the early stage of CBF recommendation systems, they were used to recommend documents such as net news [24], web pages [34] and books [30]. Both user profiles and item profiles are represented by vectors of associated weights given to a set of keywords, which are extracted from documents using information retrieval techniques [3,39] or information filtering techniques [5]. A similarity score is computed based on both profiles using a heuristic function, such as the Pearson correlation coefficient, cosine similarity, or distance-based similarity [4,24]. These systems, however, have a few limitations such as insufficient features problem [42], over-specialization problem [1], and new or unusual user problem [1,6].

On the other hand, CF-based recommendation systems typically (1) construct a user profile from rating information of each user on items, (2) select neighbors of a target user using similarity function based on the rating information, (3) predict the ratings of the target user on target items as an average, weighted sum or adjusted weighted sum of ratings given to them by neighbors, and (4) recommend top-rated *n* items. These methods of rating prediction which recommend items based on the similarity between users are called *memory-based* CF [16,31,43,46,48,49]. Another method of rating prediction is *model-based* CF in which a model such as a probabilistic model or a machine learning model is built from a large collection of ratings in order to predict ratings of target items [10,12,13,20,21,22,29,33,37,41,45].

Since the term 'CF' was first coined, various recommendation systems have been developed, i.e., *Tapestry* for recommending news articles [11], *GroupLens* for net news [35], and *Ringo* for music [42]. CF-based recommendation systems, however, also have some fundamental limitations, including new user problem [18,32], new item problem [4,26] and sparsity problem [15,18,25,32].

More recently, a number of researches have attempted to improve the quality of recommendation systems by overcoming the limitations of CBF and CF-based recommendation systems. *Fab* system [4] combines CF with CBF to lessen some weaknesses of CBF (e.g., insufficient features and over-specialization problems) and a weakness of CF (e.g., new item problem). In this system, content-based user profiles are maintained to determine similar users for collaborative recommendation. Items are recommended to a target user when the following two conditions are simultaneously satisfied: (1) each item must have a high score against the target user's profile and (2) each item should be highly rated by users whose profiles are similar to the target user' profile.

Liu et al. [28] proposed another hybrid recommendation system to deal with the problems of CBF and CF, such as sparsity and scalability. To resolve the sparsity problem, blanks in the user–item rating matrix are first filled with a weighted average rating on items which a user has rated, where the weight of the rated item is calculated by the similarity between an unrated item and the rated item in their feature values; then, CF is applied to the user–item rating matrix. Also, to resolve the scalability problem, all users are classified into different groups with respect to the user personality features. Subsequently, neighbors of a target user are found within the group to which the user belongs instead of searching the entire user space.

Yang et al. [46] proposed a new similarity function in CF techniques to address the following drawbacks. CF technique (1) is sometimes overly confident, (2) tends to discard some useful information in user profile, and (3) often derives some untrustworthy inferences when making a prediction. In their study, the similarity between a target item and each of the co-rated items are considered to determine whether or not the two items belong to the same genre of interest. When calculating the similarity between two users, Yang et al. [46] classified the co-rated items into three classes according to the differences between the ratings of the two users on the items; equal weight to all items in the same class were given as well.

Koren [20] recognized that user bias, item bias, and user preference change over time, and built, accordingly, a recommendation system by incorporating temporal information into factor modeling and item–item neighbor modeling. In another study, Koren [21] proposed a new neighbor model, where neighbor relations are modeled by minimizing the regularized squared error function. In addition, he extended the model to exploit both explicit (i.e., rating information) and implicit feedback (i.e., binary information (rated vs. not rated)) from users.

Salakhutdinov and Srebro [37] introduced a weighted version of trace-norm regularization. The trace-norm regularization is a popular method for completing the user–item rating matrix in CF. However, the method does not perform well when entries of the user–item rating matrix are sampled non-uniformly. In order to solve the problem, they proposed a trace-norm weighted by the frequency of users and items as a regularizer. Takács et al. [45] proposed several matrix factorization (MF) – based methods (i.e., a regularized MF, a fast semi-positive MF, an accurate momentum-based MF, an incremental variant of MF, and a transductive version of MF). In addition, they outlined a neighbor correction method for MF.

Shambour and Lu [40] explored the application of recommender systems in the context of the e-government domain, and proposed trustworthy business partner recommendation e-services for small-to-medium businesses. That is, they developed (1) an implicit trust filtering recommendation approach incorporating trust propagation and Jaccard metric and (2) a user-based CF recommendation approach enhanced by Jaccard metric. In addition, to further take the advantages of the two approaches, they developed a hybrid trust-enhanced CF recommendation approach (TeCF) which integrates both the approaches.

Yin and Peng [47] provided a fair framework for the performance comparison between a new recommendation algorithm and the representative/well-accepted algorithms. To that end, they presented a comparative evaluation of eight CF algorithms in detail (i.e., *k*-nearest neighbor (KNN), singular value decomposition

(SVD), non-negative MF (NMF), weighted non-negative MF (WNMF), principal component analysis (PCA) – KNN, SVD–KNN, NMF–KNN, and Eigentaste) on two datasets (i.e., Jester and MovieLens) by using three quality metrics (i.e., root mean square error (RMSE), recall, and normalized distance-based performance measure (NDPM)).

3. A proposed recommendation system

This section first provides descriptions of the notations used in our equations, which define similarity functions of traditional CF and new similarity functions proposed in this paper. Next, this section explains how we can identify neighbors of a target user using the new similarity function.

3.1. Notations

Table 1 lists the descriptions of notations used in the equations below. However, explanations on each notation will be provided for equations in Section 3.2 for the better readability of the paper.

3.2. New similarity function for selecting neighbors for each target item

This section provides a detailed explanation of a new similarity function for realizing our idea of taking into account item similarity when calculating the neighbors of a target user.

As mentioned earlier, most CF-based recommendation systems have used similarity functions to compute the similarity between two users, such as the Pearson correlation coefficient, cosine similarity, or distance-based similarity. They are defined as follows.

$$Pearson(a,b) = \frac{\sum_{i=1}^{m}(R_{a,i}-\overline{R}_a)(R_{b,i}-\overline{R}_b)}{\sqrt{\sum_{i=1}^{m}(R_{a,i}-\overline{R}_a)^2}\sqrt{\sum_{i=1}^{m}(R_{b,i}-\overline{R}_b)^2}} \quad (1)$$

$$Cosine(a,b) = \frac{\sum_{i=1}^{m}(R_{a,i})(R_{b,i})}{\sqrt{\sum_{i=1}^{m}(R_{a,i})^2}\sqrt{\sum_{i=1}^{m}(R_{b,i})^2}} \quad (2)$$

$$Distance(a,b) = \frac{1}{1+\sqrt{\sum_{i=1}^{m}(R_{a,i}-R_{b,i})^2}}, \quad (3)$$

where $R_{a,i}$, $R_{b,i}$, $\overline{R}_a, \overline{R}_b$ denote the ratings of users a and b on item i, the averages of all $R_{a,i}$ and all $R_{b,i}$, respectively; m denotes the number of items commonly purchased by both users.

These similarity functions do not consider the similarity between a target item and the other item, as observed in Eqs. (1)–(3). As such, the neighbors of a target user are computed independently of a target item. However, as we have explained in the introduction section, we need to consider the similarity between a target item and the other item when identifying the neighbors of a target user. Then, the target user's neighbors will be different for each different target item. The fact that traditional CF does not consider item similarity when computing neighbors of a target user can result in less accurate recommendations. We again argue that to make more accurate recommendations, similarity between a target item and each of the co-rated items should be considered when finding neighbors of a target user.

In addition, each of these functions calculates the similarity between a target user and every other user to find neighbors of the target user, but each from a different perspective. The Pearson correlation coefficient calculates the similarity from the perspective of *rating pattern* [2,23,26,36,38] (see Fig. 1a), cosine similarity from the perspective of *angle* between two rating vectors [2,15,26,44] (see Fig. 1b), and distance measure from the perspective of *magnitude* of rating [17,19,32] (see Fig. 1c). Depending on the similarity functions utilized, the set of like-minded users could be different and thus, items finally recommended could also differ. In light of this fact, the choice of similarity functions should be properly made based on the dataset at hand.

Therefore, when calculating similarity between users, we propose new similarity functions which adapt the above three similarity functions by giving more weight to items that is similar to a target item instead of giving equal weight to all items. For this calculation, we considered the similarity between the target item and each of the co-rated items rated by both the target user and the other user, and used the outcome as a weight. Next, Eqs. (1)–(3) are adjusted to adopt the item similarity, as in the following Eqs. (4)–(6), respectively.[1]

$$Pearson^i(a,b) = \frac{\sum_{j=1}^{m}\{Isim(i,j)^2 \times (R_{a}j-\overline{R}_a)\times(R_{b,j}-\overline{R}_b)\}}{\sqrt{\sum_{j=1}^{m}\{Isim(i,j)\times(R_{a,j}-\overline{R}_a)\}^2}\times\sqrt{\sum_{j=1}^{m}\{Isim(i,j)\times(R_{b,j}-\overline{R}_b)\}^2}} \quad (4)$$

$$Cosine^i(a,b) = \frac{\sum_{j=1}^{m}\{Isim(i,j)^2\times(R_{a,j})\times(R_{b,j})\}}{\sqrt{\sum_{j=1}^{m}\{Isim(i,j)\times(R_{a,j})\}^2}\times\sqrt{\sum_{j=1}^{m}\{Isim(i,j)\times(R_{b,j})\}^2}} \quad (5)$$

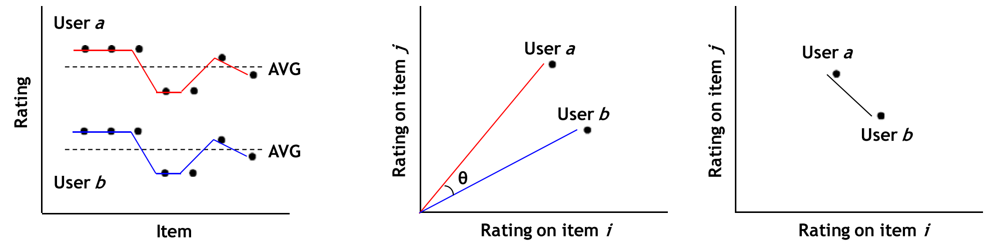
$$Distance^i(a,b) = \frac{1}{1+\sqrt{\sum_{j=1}^{m}\{Isim(i,j)\times(R_{a,j}-R_{b,j})\}^2}}, \quad (6)$$

where $Pearson^i(a,b)$, $Cosine^i(a,b)$, and $Distance^i(a,b)$ denote Pearson correlation coefficient, cosine similarity, and distance-based similarity between user a and b for target item i, respectively, and $Isim(i,j)$ denotes the similarity between target item i and other item j. $Pearson^i(a,b)$ will take any value between +1.0 and −1.0, and $Cosine^i(a,b)$ and $Distance^i(a,b)$ between 0 and 1, inclusive. Similarly to the case of calculating the similarity between two users, we

Table 1
Notations.

Notations	Descriptions
U	The number of total users
I	The number of total items
$R_{a,i}$	Rating of user a on item i
$\overline{R}_a$	average of all $R_{a,i}$ $\forall$ $i \in I$
$\overline{R}_i$	The average of all $R_{a,i}$ $\forall$ $a \in U$
m	The number of items commonly purchased by both users
u	The number of users who purchased both items i and j
$Pearson(a,b)$	Pearson correlation coefficient between users a and b
$Cosine(a,b)$	Cosine similarity between users a and b
$Distance(a,b)$	Distance-based similarity between users a and b
$Pearson^i(a,b)$	Pearson correlation coefficient between users a and b for target item i
$Cosine^i(a,b)$	Cosine similarity between users a and b for target item i
$Distance^i(a,b)$	Distance-based similarity between users a and b for target item i
$P_{a,i}^{Predicted}$	The predicted preference of target user a on target item i
$Isim(i,j)$	The similarity between items i and j
$Usim(a,b)$	The similarity between users a and b
k	The number of neighbors selected
n	The number of items recommended

[1] When we calculate the similarity between users using one of our proposed similarity functions (i.e., Eqs. (4)–(6)), the computational complexity is $O(KM^2) + O(KN)$, where M denotes the number of users, N denotes the number of items, and K denotes the number of target items. Although our proposed similarity functions require more computational complexity than that of traditional *CF* (i.e., $O(M^2)$), it is usual that the user similarity is computed offline to lessen the computational complexity problem.



(a) Pearson correlation coefficient (b) Cosine similarity (c) Distance measure

Fig. 1. Different perspectives of three similarity functions.

calculated $Isim(i,j)$ using one of the Pearson correlation coefficient, cosine similarity, and distance-based similarity as follows:

$$Pearson(i,j) = \frac{\sum_{a=1}^{u}(R_{a,i}-\overline{R}_i)\times(R_{a,j}-\overline{R}_j)}{\sqrt{\sum_{a=1}^{u}(R_{a,i}-\overline{R}_i)^2}\times\sqrt{\sum_{a=1}^{u}(R_{a,j}-\overline{R}_j)^2}} \quad (7)$$

$$Cosine(i,j) = \frac{\sum_{a=1}^{u}(R_{a,i})\times(R_{a,j})}{\sqrt{\sum_{a=1}^{u}(R_{a,i})^2}\times\sqrt{\sum_{a=1}^{u}(R_{a,j})^2}} \quad (8)$$

$$Distance(i,j) = \frac{1}{1+\sqrt{\sum_{a=1}^{u}(R_{a,i}-R_{a,j})^2}}, \quad (9)$$

where $\overline{R}_i$ and $\overline{R}_j$ denote the averages of all $R_{a,i}$ and all $R_{a,j}$, respectively, and u denotes the number of users who purchased both items i and j. As a consequence, we propose nine similarity functions which take a target item into account when computing neighbors of a target user (i.e. three user similarity functions and three item similarity functions).

We will illustrate our idea with an example. Suppose that the ratings of a target user T and other users A, B, C, D and E on five co-rated items and two target items are given, as shown in Table 2. Based on the ratings of the target user and other users on co-rated items, the similarity between target user T and each other user is calculated. Then from the ratings of selected neighbors for each target item, ratings of the target user T on the target items are predicted. In Table 3, the similarity between target user T and each other user is calculated using both the traditional similarity function (Eq. (2)) and the proposed similarity function (Eqs. (5) and (8)). If we consider only the top two users as neighbors of the target user, they are users A and B for the traditional CF, while they are users A and E for target item 6, and users B and D for target item 7 for the proposed CF. (*Note*: cosine similarity is used in this example.) Such results occur because in traditional similarity functions the ratings of users on all co-rated items are considered equally important for all target items. However, in our proposed similarity function, the ratings of users on items 1 and 2 are considered more significant for target item 6, and items 3 and 5 are more important for item 7, according to their similarity, as presented in Table 4.

3.3. Predicting preference on target items

Now, we will describe how the preference of the target user on each target item is calculated. First, all the similarities between a target user and every other user who purchased a target item are calculated. Then, top k users with the highest similarity are selected as neighbors of the target user for the target item. To predict preferences of the target user on each target item, we use the aggregation function, shown in Eq. (10), which is the function generally used by CF-based recommendation systems.

$$P_{a,i}^{Predicted} = \overline{R}_a + \frac{1}{\sum_{b=1}^{k}|Usim(a,b)|}\times\sum_{b=1}^{k}Usim(a,b)\times(R_{b,i}-\overline{R}_b) \quad (10)$$

where $P_{a,i}^{Predicted}$ denotes the predicted preference of target user a on target item i, k the number of user a's neighbors, $\overline{R}_a$ the average rating of user a, and $Usim(a,b)$ the similarity between users a and b defined in Eqs. (4)–(6).

After calculating the predicted preferences of the target user on target items via the above method, the top n items with the highest preferences are recommended for the target user.

4. Experiments

This section explains the experimental design for evaluating our ideas proposed in Section 3 as well as how the ideas affect the quality of recommendation by analyzing the results from experiments. Further, it describes the implication of the experiments.

4.1. Experimental design

In order to evaluate the effectiveness of our approach, we used two datasets. MovieLens dataset, collected from April 25, 2000 to February 24, 2003 (34 months), consists of 515,088 rating records by 1000 users on 3646 movies. Netflix dataset, collected from November 11, 1999 to December 31, 2005 (73 months), consists of 1,000,000 rating records by 9394 users on 4693 movies. The rating scale of both datasets is from 1 to 5.

Among the entire users in each dataset, 20 users who had watched movies more than 20 times via MovieLens and Netflix were randomly selected as target users. Seventy percent of each

Table 2
Example of user–item rating matrix.

User	Co-rated items					Target items	
	Item 1 (horror)	Item 2 (horror)	Item 3 (comedy)	Item 4 (drama)	Item 5 (comedy)	Item 6 (horror)	Item 7 (comedy)
A	4	4	2	3	2	5	2
B	3	3	4	3	5	3	5
C	2	2	5	3	4	2	5
D	2	3	3	3	3	3	3
E	5	4	2	3	2	5	2
T (target user)	5	4	4	3	4	-	-

Table 3
Example of similarity and rank, depending on similarity function.

T (target user)	Traditional CF		Proposed CF (Item 6)		Proposed CF (Item 7)	
	Similarity	Rank	Similarity	Rank	Similarity	Rank
T–A	0.9623	2	0.9686	1	0.9567	3
T–B	0.9642	1	0.9633	3	0.9681	1
T–C	0.9135	5	0.9060	5	0.9270	5
T–D	0.9603	3	0.9544	4	0.9674	2
T–E	0.9570	4	0.9660	2	0.9500	4

Table 4
Example of similarity between a target item and other item.

Target item 6	Similarity	Target item 7	Similarity
Item 6–Item 1	0.9904	Item 7–Item 1	0.7860
Item 6–Item 2	0.9943	Item 7–Item 2	0.8313
Item 6–Item 3	0.7892	Item 7–Item 3	0.9946
Item 6–Item 4	0.9487	Item 7–Item 4	0.9288
Item 6–Item 5	0.8047	Item 7–Item 5	0.9946

target user's data were used as training data, and the remaining data were used as test data.

In order to show that selecting neighbors depending on the target items actually improve the quality of the recommendation system, we implemented quite a few recommendation systems. We classified the systems into the following four groups. The first group includes systems using the traditional CF method and the remaining three groups include systems using the proposed CF method.

A. Three traditional CF-based recommendation systems, CF_P, CF_C and CF_D: Each of these systems computes user similarity using Pearson correlation coefficient, cosine similarity, and distance-based similarity, respectively.
B. Three Pearson-based proposed systems, CF_P_P, CF_P_C, and CF_P_D: Each of these systems computes user similarity using Pearson correlation coefficient which is weighted by item similarity using Pearson correlation coefficient, cosine similarity, and distance-based similarity, respectively.
C. Three cosine-based proposed systems, CF_C_P, CF_C_C, and CF_C_D: Each of these systems computes user similarity using cosine similarity which is weighted by item similarity using Pearson correlation coefficient, cosine similarity, and distance-based similarity, respectively.
D. Three distance-based proposed systems, CF_D_P, CF_D_C, and CF_D_D: Each of these systems computes user similarity using distance-based similarity which is weighted by item similarity using Pearson correlation coefficient, cosine similarity, and distance-based similarity, respectively.

We conducted our experiments after setting the number of neighbors to 10 and the number of recommended movies to 10, 20, 30, 40 or 50. We also utilized *precision*, *recall*, *F1*, *mean absolute error* (MAE), and *coverage* as evaluation measures. These measures have been widely used in other studies to measure and compare the quality of recommendation systems [2,7,9,14,17,23,27,32,36]. All benchmark systems in group A and our proposed recommendation systems in group B, C, and D were implemented using Transact-SQL in Microsoft SQL Server 2008.

4.2. Experimental results and analysis

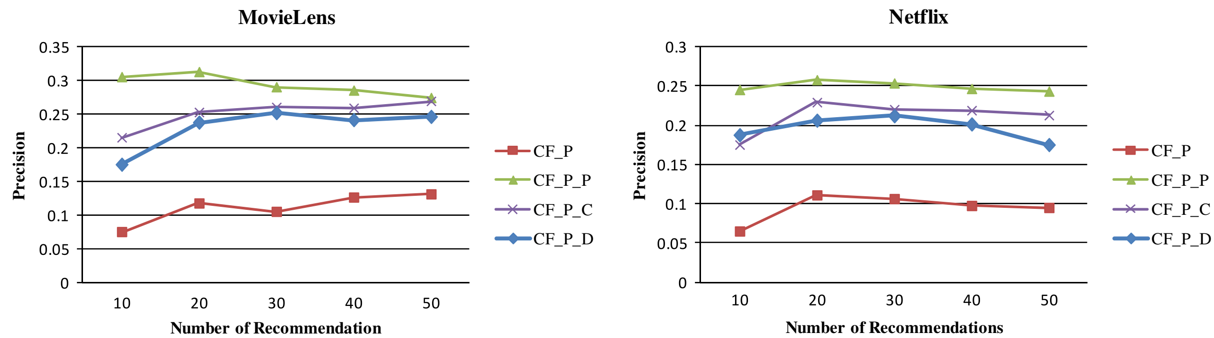
We compared three benchmark systems in group A with systems in group B, group C, and in group D, respectively. First, all three systems in group B (i.e., CF_P_P, CF_P_C, and CF_P_D) revealed considerably better quality of recommendation than the benchmark system using the Pearson correlation coefficient (i.e., CF_P) in group A, regardless of the number of recommendations in precision, recall, and F1 in both datasets (see Fig. 2a–c). Among the systems in groups A and B, the best precision was achieved by CF_P_P when the number of recommendations was 20, and the best recall and F1 was accomplished when the number of recommendations was 50. Similar experiments were conducted to compare systems in group C and D with systems in group A, respectively. In the former case, the best precision was achieved by CF_C when the number of recommendations was 20, and the best recall and F1 were achieved by CF_C_P when the number of recommendations was 50, in both datasets. In the latter case, the best precision was achieved by CF_D_P when the number of recommendations was 20, and the best recall and F1 when the number of recommendations was 50, in both datasets. In order to maintain simplicity and readability of the paper, we only present the best results from each dataset in Fig. 2.

When computing user similarity, systems in group B (i.e., CF_P_P/C/D) showed better quality of recommendation than the systems in group C and group D which use other similarity functions (i.e., CF_C/D_P/C/D) in all cases. (*Note*: A/B means A or B.) When computing the item similarity, systems which calculate item similarity using Pearson correlation coefficient (i.e., CF_P/C/D_P) demonstrated better quality of recommendation than systems which calculate item similarity using other similarity functions (i.e., CF_P/C/D_C/D) in most cases. Among the three traditional systems and our nine proposed systems, CF_P_P revealed the best quality of recommendation.

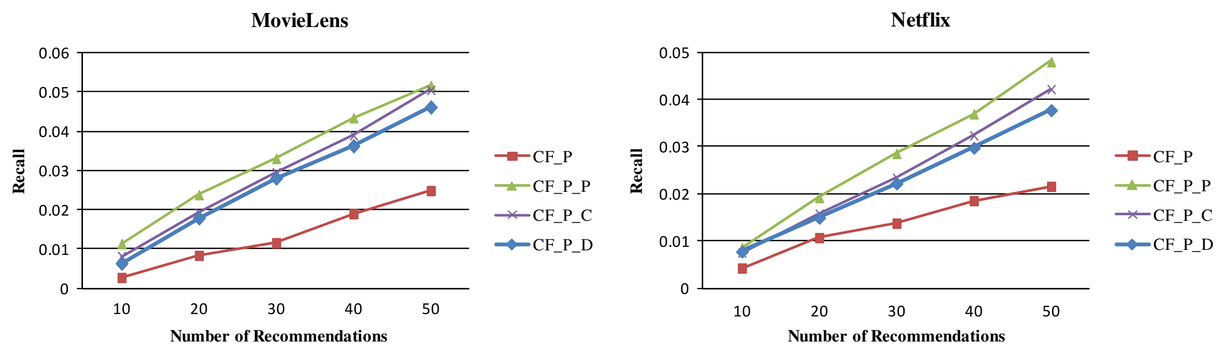
In MAE, all our proposed systems in group B, C, and D demonstrated better quality of recommendation than their corresponding traditional benchmark systems in group A, as shown in Fig. 3. The results from both datasets were similar to each other. However, the effect of our approach on error reduction was somewhat different from each other. In the MovieLens dataset, the lowest MAE was achieved by CF_P_D, which was also lower than that of rational inference based algorithm (IB) proposed in another study [46], which is similar to our system in that they also considered the similarity between items. Our study, however, is different from their study in the following aspect. They assigned an equal weight to items belonging to the same class, whereas we assigned different weights to individual items. Thus, we were able to identify neighbors who are more appropriate to each target item than those identified by their approach. In the Netflix dataset, the lowest MAE was achieved by CF_P_C. Overall, systems which calculate user similarity based on Pearson correlation coefficient reflected better performance than systems which calculate user similarity based on other similarity functions.

Since coverage means how many items can be considered as candidate items to recommend, diversity of items to recommend for a target user tends to increase as the coverage increases. As shown in Fig. 4, our proposed systems also highly outperformed the traditional systems in coverage in all cases. The best coverage was achieved when cosine similarity and distance-based similarity were used to calculate item similarity in both datasets.

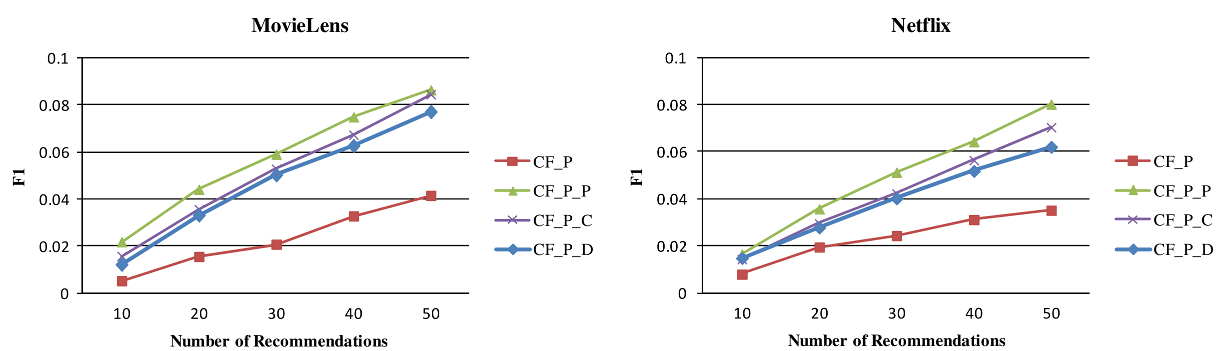
Throughout the experiments on the above two datasets, the systems in group B exhibited better performance than systems in



(a) Precision



(b) Recall



(c) F1

Fig. 2. Comparison between CF_P (a traditional system using Pearson correlation coefficient) and CF_P_P/C/D (systems using Pearson correlation coefficient and weighted by item similarity functions).

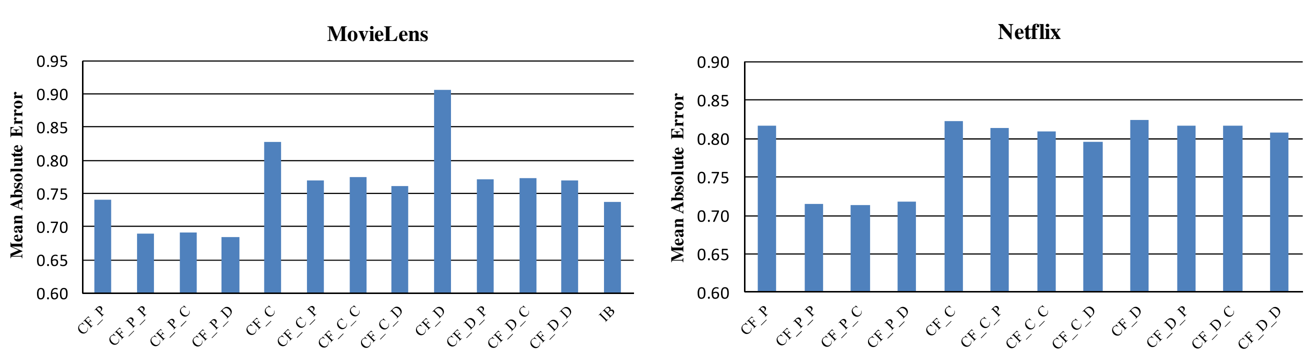


Fig. 3. Comparison in MAE between traditional systems and systems that implement our idea.

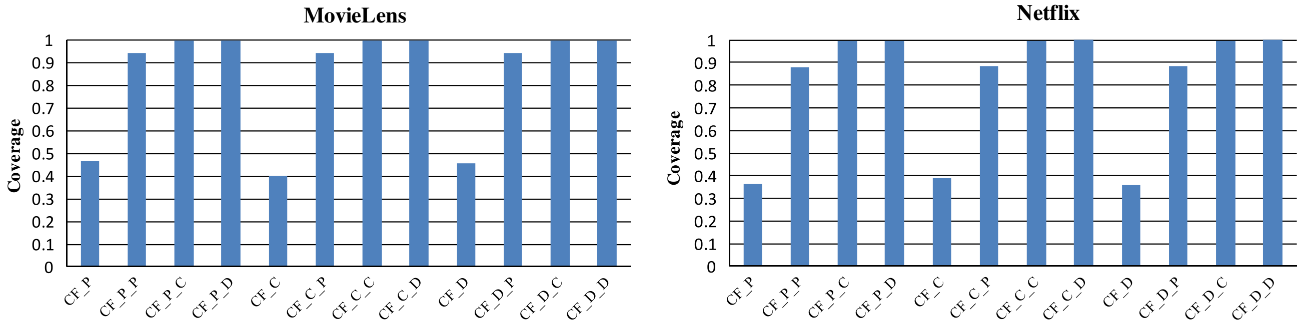


Fig. 4. Comparison in coverage between traditional systems and systems that implement our idea.

other groups in all measures used in this study. Through the results of our experiments, our approach proved to be effective in selecting neighbors for each target item and also in improving the quality of CF-based recommendation system in precision, recall, F1, MAE, and coverage.

5. Conclusions

In traditional CF-based recommendation systems, neighbors of a target user are identical for all target items since the systems do not consider a target item. However, it should not be neglected that neighbors of a target user should be selected depending on individual target items. Therefore, similarity between a target item and each of the co-rated items which both users rate, should be considered when finding neighbors of a target user.

This study proposed a new similarity function to improve the quality of recommendation based on the idea of selecting different neighbors of a target user for each different target item. This idea was implemented as follows. We first calculated the item similarity between the target item and each of the co-rated items, and then used it as a weight of each co-rated item when calculating the user similarity between the target user and every other user.

We conducted a series of experiments to compare traditional recommender models with our proposed recommender models. According to the results of the experiments, our proposed recommender models considerably outperformed traditional recommender models in precision, recall, F1, MAE, and coverage, especially when the user similarity is calculated using the Pearson correlation coefficient which is weighted by item similarity. From the experiments, we conclude that our idea of selecting different neighbors of a target user for each different target item contributes to recommending more accurate and diverse items which target users want. More specifically, our system results in the improvement of recommendation quality of CF-based techniques. Moreover, when finding neighbors, our recommender model assigns different weights to individual items, and makes more accurate recommendations than a recommender model that gives equal weight to items belonging to the same class, as conducted in the study by Yang et al. [46]. Therefore, our study allows us to ensure that it is important to define the similarity function for identifying neighbors of a target user based on the most appropriate information possible.

In this study, we have conducted experiments with a fixed number of neighbors. However, it may be worthwhile to perform experiments by changing the number of neighbors, although similar results are expected.

References

[1] G. Adomavicius, A. Tuzhilin, Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (2005) 734–749.
[2] A. Albadvi, M. Shahbazi, A hybrid recommendation technique based on product category attributes, Expert Systems with Applications 36 (2009) 11480–11488.
[3] R.A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, 1999.
[4] M. Balabanovic, Y. Shoham, Content-based, collaborative recommendation, Communications of the ACM 40 (1998) 66–72.
[5] N.J. Belkin, W.B. Croft, Information filtering and information retrieval: two sides of the same coin, Communications of the ACM 35 (1992) 29–38.
[6] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Proceedings of the 15th International Conference on Machine Learning, 1998.
[7] J. Bobadilla, F. Ortega, A. Hernando, J. Alcala, Improving collaborative filtering recommender system results and performance using genetic algorithms, Knowledge-Based Systems 24 (2011) 1310–1316.
[8] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, Knowledge-Based Systems 23 (2010) 520–528.
[9] L.S. Chen, F.H. Hsu, M.C. Chen, Y.C. Hsu, Developing recommender systems with the consideration of product profitability for sellers, Information Sciences 178 (2008) 1032–1048.
[10] K.W. Cheung, J.T. Kwok, M.H. Law, K.C. Tsui, Mining customer product ratings for personalized marketing, Decision Support Systems 35 (2003) 231–243.
[11] D. Goldberg, D. Nichols, B.M. Oki, D. Terry, Using collaborative filtering to weave an information tapestry, Communications of the ACM 35 (1992) 61–70.
[12] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Information Retrieval 4 (2001) 133–151.
[13] T. Hofmann, Latent semantic models for collaborative filtering, ACM Transactions on Information Systems 22 (2004) 89–115.
[14] C.L. Huang, W.L. Huang, Handling sequential pattern decay: developing a two-stage collaborative recommendation system, Electronic Commerce Research and Applications 8 (2009) 117–129.
[15] B. Jeong, J. Lee, H. Cho, An iterative semi-explicit rating method for building collaborative recommender systems, Expert Systems with Applications 36 (2009) 6181–6186.
[16] D. Joaquin, I. Naohiro, Memory-based weighted-majority prediction, in: Proceedings of ACM SIGIR' 99 Workshop on Recommender Systems: Algorithms and Evaluation, 1999.
[17] H.K. Kim, J.K. Kim, Y.U. Ryu, Personalized recommendation over a customer network for ubiquitous shopping, IEEE Transactions on Services Computing 2 (2009) 140–151.
[18] H.N. Kim, A.T. Ji, I. Ha, G.S. Jo, Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation, Electronic Commerce Research and Applications 9 (2010) 73–83.
[19] J.K. Kim, H.K. Kim, Y.H. Cho, A user-oriented contents recommendation system in peer-to-peer architecture, Expert Systems with Applications 34 (2008) 300–312.
[20] Y. Koren, Collaborative filtering with temporal dynamics, Communications of the ACM 53 (2010) 89–97.
[21] Y. Koren, Factor in the neighbors: scalable and accurate collaborative filtering, ACM Transactions on Knowledge Discovery from Data 4 (2010) 1–24.
[22] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Recommender systems: a probabilistic analysis, Journal of Computer and System Science 63 (2001) 42–61.
[23] K. Kwon, J. Cho, Y. Park, Multidimensional credibility model for neighbor selection in collaborative recommendation, Expert Systems with Applications 36 (2009) 7114–7122.
[24] K. Lang, NewsWeeder: learning to filter netnews, in: Proceedings of the 12th International Conference on Machine Learning, 1995.
[25] J.S. Lee, S. Olafsson, Two-way cooperative prediction for collaborative filtering recommendations, Expert Systems with Applications 36 (2009) 5353–5361.
[26] T.Q. Lee, Y. Park, Y.T. Park, A time-based approach to effective recommender systems using implicit feedback, Expert Systems with Applications 34 (2008) 3055–3062.
[27] D.R. Liu, C.H. Lai, W.J. Lee, A hybrid of sequential rules and collaborative filtering for product recommendation, Information Sciences 179 (2009) 3505–3519.
[28] Z. Liu, W. Qu, H. Li, C. Xie, A hybrid collaborative filtering recommendation mechanism for P2P networks, Future Generation Computer Systems 26 (2010) 1409–1417.

[29] B. Marlin, Modeling user rating profiles for collborative filtering, Advances in Neural Information Processing Systems 16 (2003) 627–634.
[30] R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, Paper presented at the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, 1999.
[31] A. Nakamura, N. Abe, Collaborative filtering using weighted majority prediction algorithm, in: Proceedings of the 15th International Conference on Machine Learning, 1998.
[32] Y.J. Park, K.N. Chang, Individual and group behavior-based customer profile model for personalized product recommendation, Expert Systems with Applications 36 (2009) 1932–1939.
[33] D.Y. Pavlov, D.M. Pennock, A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains, Advances in Neural Information Processing Systems 15 (2002) 1441–1448.
[34] M. Pazzani, D. Billsus, Learning and revising user profile: the identification of interesting web sites, Machine Learning 27 (1997) 313–331.
[35] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, 1994.
[36] S. Russell, V. Yoon, Applications of wavelet data reduction in a recommender system, Expert Systems with Applications 34 (2008) 2316–2325.
[37] R. Salakhutdinov, N. Srebro, Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm, 2010, pp. 1–9. arXiv:1002.2780v.
[38] J. Salter, N. Antonopoulos, CinemaScreen recommender agent: combining collaborative and content-based filtering, IEEE Intelligent Systems 21 (2006) 35–41.
[39] G. Salton, Automatic Text Processing, Addison-Wesley Longman Publishing Co., Inc., Boston, 1988.
[40] Q. Shambour, J. Lu, A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services, International Journal of Intelligent Systems 26 (2011) 814–843.
[41] G. Shani, D. Heckerman, R.I. Brafman, An MDP-based recommender system, Journal of Machine Learning Research 6 (2002) 1265–1295.
[42] U. Shardanand, P. Maes, Social information filtering algorithms for automating "Word of Mouth", in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1995.
[43] L. Si, R. Jin, Flexible mixture model for collaborative filtering, in: Proceedings of 20th International Conference on Machine Learning, 2003.
[44] P. Symeonidis, A. Nanopoulos, Y. Manolopoulos, Providing justifications in recommender systems, IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems an Humans 38 (2008) 1262–1272.
[45] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, Journal of Machine Learning Research 10 (2009) 623–656.
[46] J.M. Yang, K.F. Li, D.F. Zhang, Recommendation based on rational inferences in collaborative filtering, Knowledge-Based Systems 22 (2009) 105–114.
[47] C.X. Yin, Q.K. Peng, A careful assessment of recommendation algorithms related to dimension reduction techniques, Knowledge-Based Systems 27 (2012) 407–423.
[48] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, H.-P. Kriegel, Probabilistic memory-based collaborative filtering, IEEE Transactions on Knowledge and Data Engineering 16 (2004) 56–69.
[49] K. Yu, X. Xu, J. Tao, M. Ester, H.-P. Kriegel, Instance selection techniques for memory-based collaborative filtering, in: Proceeding of the Second SIAM International Conference on Data Mining (SDM' 02), 2002.