Extended Collaborative Filtering Technique for Mitigating the Sparsity Problem

K. Choi, Y. Suh, D. Yoo

Keunho Choi
Korea Workers' Compensation and Welfare Service
8, Beodeunaru-ro 2-gil, Yeongdeungpo-gu, Seoul, Republic of Korea
ckh0515@hanmail.net

Yongmoo Suh
Korea University Business School
Anam-Ro 145, Seongbuk-Gu, Seoul, Republic of Korea
ymsuh@korea.ac.kr

Donghee Yoo*
Department of Management Information Systems, Gyeongsang National University, BERI
501 Jinju-daero, Jinju, Republic of Korea
*Corresponding author: dhyoo@gnu.ac.kr

Abstract: Many online shopping malls have implemented personalized recommendation systems to improve customer retention in the age of high competition and information overload. Sellers make use of these recommendation systems to survive high competition and buyers utilize them to find proper product information for their own needs. However, transaction data of most online shopping malls prevent us from using collaborative filtering (CF) technique to recommend products, for the following two reasons: 1) explicit rating information is rarely available in the transaction data; 2) the sparsity problem usually occurs in the data, which makes it difficult to identify reliable neighbors, resulting in less effective recommendations. Therefore, this paper first suggests a means to derive implicit rating information from the transaction data of an online shopping mall and then proposes a new user similarity function to mitigate the sparsity problem. The new user similarity function computes the user similarity of two users if they rated similar items, while the user similarity function of traditional CF technique computes it only if they rated common items. Results from several experiments using an online shopping mall dataset in Korea demonstrate that our approach significantly outperforms the traditional CF technique.
Keywords: recommendation system, collaborative filtering, sparsity problem, similarity function.

1 Introduction

In the age of information overload, information about products and services on the Internet is growing explosively; consequently, people have difficulty in processing the overwhelming amount of information that is available. To address this problem, a number of personalized recommendation techniques have been introduced by many studies, so that sellers of online shopping malls can survive high competition and buyers can locate the best product information for their own needs [1–3]. Personalized recommendations are usually seen as a specific kind of information filtering that enables people to filter out unnecessary and uninteresting information [4]. Among the many recommendation techniques that have been suggested, collaborative filtering (CF) has been widely adopted in many practical applications due to its simplicity and effectiveness and has proven to be useful [2].

However, it is still not easy for most online shopping malls to make use of the collaborative filtering technique for recommendation, because explicit rating information required by the technique is rarely available in online shopping malls and/or because there is a high chance of sparsity in the transaction data of online shopping malls. Thus, there is a need to devise a way to derive implicit rating information that can play the role of explicit rating information and extend the collaborative filtering technique so that it can be used effectively even when there is a sparsity problem in the transaction data of online shopping malls.

It is known that personalized recommendation and improved customer retention forms a virtuous cycle in which good quality of recommendation leads to improved customer retention which in turn leads to the better quality of recommendation through more customer input into the recommendation system [2]. As such, the recommendation system that we propose in this paper will contribute to the higher level of customer retention of online shopping malls, which will help them to survive today's high competition.

In this paper, therefore, we first suggest a means to derive implicit rating information from the transaction data available on an online shopping mall. We assumed that the number of purchasing the same item represents the preference toward the item and thus can be used as implicit rating information on the item. Then we propose a new user similarity function that can mitigate the sparsity problem. The traditional collaborative filtering technique calculates the user similarity of two users only when they have rated common items. In that case, we come across bad recommendation quality due to the sparsity problem. Instead, our new user similarity function computes user similarity if they rated similar items which include common items. So, we define an item similarity function as a function that computes the item similarity of all pairs of items purchased by two users whose similarity is to be computed and use the item similarity to define the user similarity function.

We have implemented both a recommendation system which uses the extended collaborative filtering technique with our new user similarity function and a benchmark system which adopts the traditional collaborative filtering technique. Results from a series of experiments using the transaction data of an online shopping mall in Korea clearly demonstrate that our approach significantly improves the quality of recommendation systems, compared with those obtained from a benchmark system.

The rest of this paper is organized as follows. Section 2 reviews the previous studies on recommendation systems, especially those which have attempted to solve the sparsity problem of the collaborative filtering technique. Section 3 describes our proposed similarity function in detail, and Section 4 provides the details of the several experiments conducted to verify our approach and the results from those experiments. The last section concludes our paper with summary, implications, and limitations.

2   Previous works

This section reviews general recommendation techniques with a focus on the collaborative filtering technique related to our study.

2.1   General recommendation techniques

The techniques used in most recent recommendation systems can generally be classified into one of the following four types: *Content-based filtering* (CBF); *Collaborative filtering* (CF); *Rule-based approach*; *Hybrid approach*.

CBF typically 1) constructs a content-based *item profile* by extracting a set of features from each item in the item set; 2) builds a content-based *user profile* from a set of features of the

items that each user purchased; 3) calculates the similarity between the user profiles and the item profiles; and 4) recommends the top n items with the highest similarity scores. In other words, CBF mainly recommends items based on the similarity between items to recommend and items already purchased [5, 6]. However, CBF has several limitations: 1) it is not easy to obtain a sufficient number of features for item profiles and user profiles (*insufficient features problem*) [7]; 2) items that can be recommended are limited to those similar to the items that a target user previously purchased (*over-specialization problem*) [8]; and 3) new users who have not yet purchased items cannot get appropriate recommendations (*new user problem*) [9].

CF typically 1) builds a rating-based user profile from the rating information of each user on items; 2) identifies neighbors (also called like-minded users) who rated items similarly as the target user; 3) predicts ratings of the target user on target items purchased not by the target user, but by the neighbors; and 4) recommends the top n items with the highest predicted ratings. However, CF also has some limitations: 1) it is difficult to recommend items for users who have not yet rated items (*new user problem*) [10, 11]; 2) it is difficult to recommend items that have never been rated by users before (*new item problem*) [5, 12]; and 3) it makes poor recommendations when rating information is insufficient (*sparsity problem*) [4, 13].

Rule-based approach typically derives rules among items in the item set from a large transaction dataset collected over time, using data mining techniques. The rules could be either *association rules* among items purchased together [14] or *sequential patterns* among items purchased in sequence over time [15, 16]. However, the rule-based approach to recommending items has limitations in that it is difficult to recommend items that do not appear in association rules or sequential patterns; moreover, it does not take into account users' preference (or rating information) on items.

Hybrid approach has been developed to overcome – or at least reduce – the weaknesses of CBF, CF, and the rule-based approach [4, 5]. In general, the hybrid approach makes recommendations by combining results from each recommendation technique, selecting one of recommendation techniques to be applied according to specific criteria, or embedding one or more recommendation techniques in applying other recommendation techniques.

2.2   Collaborative filtering

Thus far, many recommendation systems using the CF technique have been developed and used in many practical applications, such as Tapestry for recommending news articles [17], GroupLens for net news [18], and Ringo for music [7]. The CF technique utilizes users' rating information on items to represent their preference on corresponding items and predicts a target user's ratings of items based on the user's similarity in ratings [19, 20].

The CF technique can be further classified into *model-based* and *memory-based* CF techniques. In the model-based CF technique, a model such as a probabilistic model or a machine learning model is built from a large collection of ratings in order to predict a target user's ratings on target items [21–23]. Koren [23] suggested the new neighbor model, in which neighbor relations were modeled by minimizing the regularized squared error function. In addition, he extended the model to utilize both explicit (i.e., rating information) and implicit feedback (i.e., binary information [rated vs. not rated]) from users. Salakhutdinov and Srebro [24] 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.

In the memory-based CF technique, items are recommended mainly based on the similarity

between users as described in Section 2.1 [25, 26]. Balabanovic and Shoham [5] developed the Fab system, which combines CF with CBF to improve the accuracy of recommendation systems by mitigating new item, insufficient features, and over-specialization problems inherent to CF and CBF. In the Fab system, items are recommended to a target user if and only if each item is highly similar to the target user's profile and each item is rated by the neighbors of the target user. Yang et al. [27] attempted to propose a new similarity function to be used in CF techniques in order to deal with the weaknesses of CF technique, namely, 1) CF is sometimes overly confident, 2) CF tends to discard some useful information in user profiles, and 3) CF often derives some untrustworthy inferences when making a prediction. To this end, they took into account the similarity between a target item and each of the co-rated items in order to determine whether the two items belong to the same genre of interest or not. In addition, they calculated the similarity between two users by giving different weights to the co-rated items classified into three classes according to the differences between the ratings of the two users on the items.

In order to alleviate the sparsity problem, Liu et al. [28] proposed a hybrid recommenda-tion system. They first filled a blank in the user-item rating matrix with a weighted average rating of items already rated by the user, where the weights of the rated items were calculated by the similarity between an unrated item and the rated items based on feature values. CF is then applied to the user-item rating matrix. Our approach is different from their approach in that our approach uses only users' ratings on items to mitigate the sparsity problem, whereas their approach needs additional information about features of items, which may cause additional problems (e.g., insufficient features), thereby reducing the application area. Shambour and Lu [13] proposed a hybrid trust-enhanced CF recommendation approach (TeCF), which integrates both an implicit trust-filtering recommendation approach and an enhanced user-based CF recom-mendation approach. By incorporating trust propagation, they attempted to relax the sparsity problem. Although this approach can extend the number of potential neighbors, the reliabil-ity of the similarity between potential neighbors still needs to be improved. Our study focuses primarily on improving the reliability of the similarity between users in order to mitigate the sparsity problem in a memory-based CF. Formoso et al. [10] proposed an approach called *profile expansion* based on the query expansion techniques used in information retrieval to mitigate the new user problem which can cause the sparsity problem. In their study, the size of a user's original profile increased by adding a set of item-rating pairs to the profile based on item-global, item-local, or user-global profile expansion technique. Item-global profile expansion technique finds items similar to the items already exist in the user profile and adds the found items to the user profile, while item-local profile expansion technique finds items to be added to the user profile based on the items recommended to the user. User-local profile expansion technique finds the user's neighbors and adds items rated by them to the user profile. The difference between their approach and our approach is that they calculate a user similarity between two users based on the expanded user profile, while we compute the user similarity by taking into account item similarity of all pairs of items rated by these users without expanding the user profile.

3 Proposed approach

This section provides the explanation on the notations used in the equations that define a new similarity function and explains how we extended ratings on only commonly rated items to ratings on all pairs of items to mitigate the sparsity problem using the new user similarity function.

Table 1: Notations

Notations	Descriptions
U	The number of total users
I	The number of total items
$AP(A,i)$	Absolute preference of user A on item i
$RP(A,i)$	Relative preference of user A on item i
$R_{A,i}$	Rating of user A on item i
m	The number of items commonly rated by both users
U_{ij}	The number of users who rated both items i and j
$Cosine(A,B)$	Cosine similarity between users A and B
$P_{A,i}^{Predicted}$	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

3.1   Notations

Table 1 shows the description of notations used in the equations that define a new user similarity function. However, the equations will be provided in Section 3.2, Section 3.3, and Section 3.4 again with explanation on each notation for the better readability of the paper.

3.2   Deriving implicit ratings of users on items

In many online shopping malls, it is usually difficult to obtain explicit rating information on items. In order to apply the CF technique to such circumstance, this study derived implicit ratings of users on items from transaction data as an alternative to explicit ratings.

First, the absolute preference of user A on item i, $AP(A,i)$, is calculated from following equation.

$$AP\left(A,i\right) \;=\; \ln\left(\frac{\text{The number of transactions of user } A \text{ including item } i}{\text{The number of transactions of user } A} + 1\right) \tag{1}$$

Since it only takes into account the frequency of purchase, the absolute preference of user A on item i increases as the number of transactions of user A including that item increases. This value, however, may not represent the preference of user A on item i exactly because the frequency of purchase is quite different depending on the item price, item lifetime, and so on. For instance, since expensive items or items with long lifespan are usually purchased infrequently, the preferences of users on them cannot be higher than those on cheap items or on items with short lifespan. Thus, it is necessary to define relative preference which is comparable among items using the absolute preferences of other users on item i. The relative preference of user A on item i, $RP(A,i)$ is thus defined as in equation (2).

$$RP\left(A,i\right) = \frac{AP(A,i)}{\text{Max}.AP(u,i)} \tag{2}$$

, where u denotes every user who purchased item i. The reason for using Max function as a denominator in equation (2) is to make $RP(A,i)$ range from 0.0 to 1.0 (i.e., normalization).

Finally, $RP(A,i)$ is multiplied by 5 and rounded up so that *implicit rating* ranges from 1 to 5, as is mostly used in current recommendation systems, which is explained by equation (3).

Table 2: Example of user-item implicit rating matrix

User	Item 1	Item 2	Item 3	Item 4	Item 5
A	-	4	2	3	-
B	3	-	4	-	5
C	2	2	-	3	-
D	-	3	-	3	3
E	5	-	-	3	2

$$Implicit\ rating\,(A,i) = Round\ up\ (5\ \times\ RP\,(A,i)) \tag{3}$$

3.3 New user similarity function for mitigating the sparsity problem

With the implicit ratings of users on items derived in Section 3.2, the similarity between a target user and every other user is calculated. As mentioned earlier, traditional CF-based recommendation systems have calculated the similarity between two users from the rating information of co-rated items by both users, as shown in equation (4). The similarity function defined in equation (4) is *cosine similarity* which is one of widely used similarity functions in CF.

$$Cosine\,(A,B) = \frac{\sum_{i=1}^{m}(R_{A,i})(R_{B,i})}{\sqrt{\sum_{i=1}^{m}(R_{A,i})^2}\sqrt{\sum_{i=1}^{m}(R_{B,i})^2}} \tag{4}$$

, where $R_{A,i}$ and $R_{B,i}$ denote the ratings of users A and B on item i, respectively and m denotes the number of items commonly rated by both users. For example, in traditional CF approach, the similarity between users A and B is calculated from the rating information of item 3 which is commonly rated by both users (see Table 2). However, this may result in untrustworthy inference in calculating the similarity between them since insufficient rating information (i.e., rating information of users A and B only on item 3) is used to calculate the similarity, which is known as the *sparsity problem*. In addition, some potentially useful information (i.e., rating information of user A on items 2 and 4, and those of B on items 1 and 5) may be discarded in traditional CF approach.

In order to mitigate the above problems, this paper adapted the cosine similarity as equation (5) by assuming that two items i and j can be regarded as *commonly rated* by two users A and B if they are similar to each other and rated by user A or user B. Using the equation (5), we can calculate the user similarity between users A and B based on the item similarity, computed by considering rating information of all pairs of *commonly rated items* (i.e., $N_{I_A} \times N_{I_B}$, where N_{I_A} and N_{I_B} denote the number of items rated by users A and B, respectively). In the example of table 2, item similarity is defined to include all pairs of *commonly rated items* by users A and B when the user similarity between users A and B is to be computed (i.e., nine item pairs: (item 2, item 1), (item 2, item 3), (item 2, item 5), (item 3, item 1), (item 3, item 3), (item 3, item 5), (item 4, item 1), (item 4, item 3), and (item 4, item 5)). Our approach, therefore, improves the reliability of the similarity between two users by utilizing rating information of all similar item pairs, thereby mitigating the sparsity problem caused by insufficient rating information. User similarity between two users A and B is defined as follows:

$$USIM\,(A,B) = \frac{\sum_{i \in I_A}\sum_{j \in I_B}\left\{ISIM\,(i,j)^2 \times (R_{A,i}) \times (R_{B,j})\right\}}{\sqrt{\sum_{i \in I_A}\sum_{j \in I_B}\left\{ISIM\,(i,j) \times (R_{A,i})\right\}^2} \times \sqrt{\sum_{i \in I_A}\sum_{j \in I_B}\left\{ISIM\,(i,j) \times (R_{B,j})\right\}^2}} \tag{5}$$

, where I_A and I_B denote a set of items rated by user A and user B, respectively, and $ISIM(i, j)$, the similarity between item i and item j, is calculated similarly to $USIM(A, B)$[1], using the cosine similarity (i.e., equation (4)), as defined in equation (6).

$$ISIM\left(i,j\right)=\frac{\sum_{A=1}^{U_{ij}}\left(R_{A,i}\right)\times\left(R_{A,j}\right)}{\sqrt{\sum_{A=1}^{U_{ij}}\left(R_{A,i}\right)^2}\times\sqrt{\sum_{A=1}^{U_{ij}}\left(R_{A,j}\right)^2}} \tag{6}$$

, where U_{ij} denotes the number of users who rated both items i and j.

3.4   Predicting preference of a target user on target items

After calculating the user similarity between a target user and every other user, the top k users with the highest similarity are selected as neighbors of the target user. Then, rating information of the neighbors is used to predict the preference of the target user on target items, as shown in equation (7).

$$P_{A,i}^{Predicted}=\frac{1}{\sum_{B=1}^{k}\left|USIM\left(A,B\right)\right|}\times\sum_{B=1}^{k}USIM\left(A,B\right)\times R_{B,i} \tag{7}$$

, where $P_{A,i}^{Predicted}$ denotes the predicted preference of target user A on item i, k the number of user A's neighbors, and $USIM(A,B)$ the similarity between the target user A and A's neighbor user B calculated using equation (5).

Finally, the top n items with the highest preference are recommended for the target user, where items already purchased by the target user may be included in recommendation list since this study assumes that users may repurchase the same item, differently from usual recommendation systems.

4   Experiments

This section describes the experimental design for evaluating the effect of our ideas proposed in Section 3 on the accuracy of recommendation, and explains the implication of the results from the experiments.

4.1   Experimental design

The data used in our experiment were provided by one of the biggest online shopping malls in Korea from August 16, 2008 to August 15, 2009 (12 months), which consists of 15,860 transactions of 234 users on 1,097 items[2], and shows high sparsity rate of 98.59%[3]. Prior to conducting our experiments, we divided our dataset into four parts, as shown in Fig 1. Firstly, it was divided by time into Part A and B, and secondly by random sampling of users into Part C and D. Part A consists of transaction data collected during the first 6 months and Part B during the second 6 months. Part C consists of transaction data from 70% of the users, randomly chosen, and Part D the transaction data of the remaining users.

---

[1] $USIM(A, B)$ takes a value between 0 and 1.

[2] Since it is difficult to recommend items to users who have purchased a small number of transactions as in most recommendation systems (i.e., new user problem), this study focused on the users who are involved in more than 30 transactions among total 1,000 users.

[3] 3,626 distinct transactions among 256,698 ($= 234 \times 1,097$) possible transactions.

Figure 1: Four parts of our dataset

When making recommendations, we used Part A*(C+D) in order to calculate the similarity between each target user and every other user, and recommended items for the target users in Part B*D. In our experiment, as a means to measure the quality of recommendation, we also used precision, recall, and F1, as used elsewhere to evaluate and compare the quality of recommendations. F1 is the harmonic average of precision and recall.

In order to ensure that our proposed approach actually improves the quality of recommendation system, we implemented both our recommendation system and a benchmark system using Transact-SQL on Microsoft SQL Server 2008.

4.2   Experimental results and analysis

In order to compare our proposed recommendation system with the benchmark system (i.e., traditional CF approach which calculates the similarity between users using equation (4)), we conducted several experiments. In the first experiment, we evaluated the effects of both the number of recommendations and the number of neighbors on the accuracy of the benchmark system. As shown in Fig. 2 (a), when the number of neighbors was low (i.e., $k \leq 40$), the precision of benchmark system tends to increase as the number of recommendations increases. On the contrary, however, when the number of neighbors was high (i.e., $k \geq 60$), the precision of benchmark system tends to increase as the number of recommendations decreases. The best precision was achieved when the number of neighbors was 120 and the number of recommendations was 10. Generally, as shown in Fig. 2 (b) and (c), the recall and F1 of benchmark system tend to increase as the number of recommendations increases, regardless of the number of neighbors. However, as the number of neighbors increases, the effects of the number of recommendations on the recall and F1 of benchmark system decrease. The best recall was achieved when the number of neighbors was 10 and the number of recommendations was 50, and the best F1 when the number of neighbors was 40 and the number of recommendations was 50.

Similar experiment was conducted to evaluate the effects of both the number of recommendations and the number of neighbors on the accuracy of our proposed system. As shown in Fig. 3 (a), when the number of neighbors was low (i.e., $k \leq 20$), the precision of proposed system tends to increase as the number of recommendations decreases. In addition, when the number of neighbors was high (i.e., $k \geq 60$), the precision of proposed system also tends to increase as the number of recommendations decreases. The best precision was achieved when both the number of neighbors and the number of recommendations were 10. Generally, as shown in Fig. 3 (b) and (c), the recall and F1 of proposed system tend to increase as the number of recommendations increases, regardless of the number of neighbors. However, as the number of neighbors increases, the effects of the number of recommendations on the recall and F1 of proposed system decrease. The best recall was achieved when the number of neighbors was 10 and the number of recommendations was 50, and the F1 when the number of neighbors 10 and the number of recommendations was 30.

In the final experiment, we compared the best precision, recall, and F1 of benchmark system
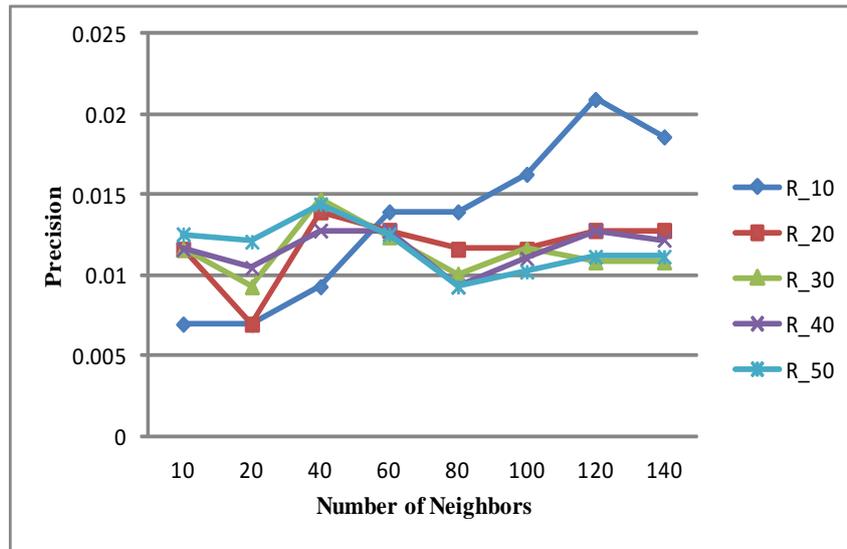
(a) Precision
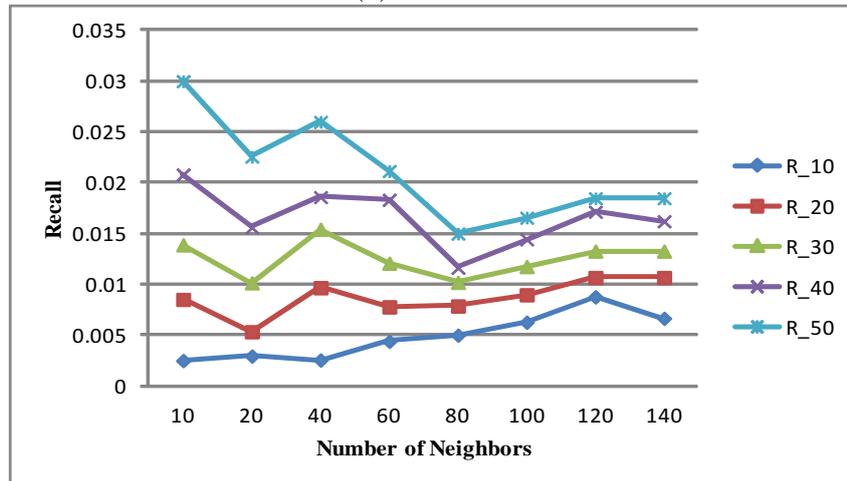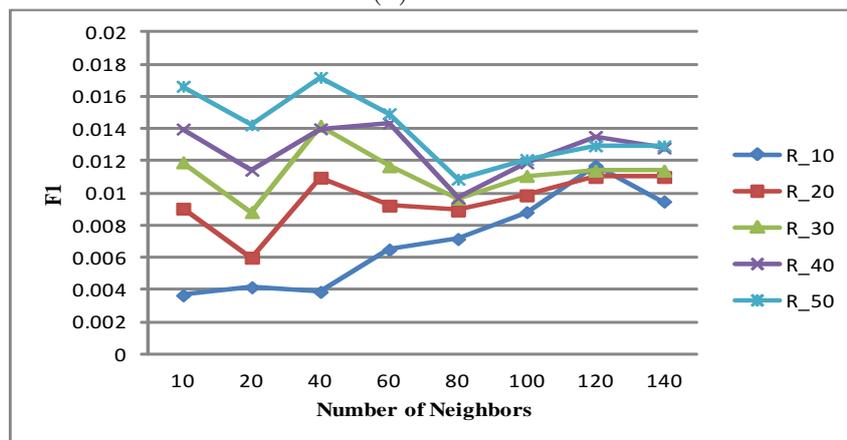
(b) Recall

(c) F1

Figure 2: The effect of both the number of recommendations and the number of neighbors on the accuracy of benchmark system (N in R_N represents the number of recommendations)
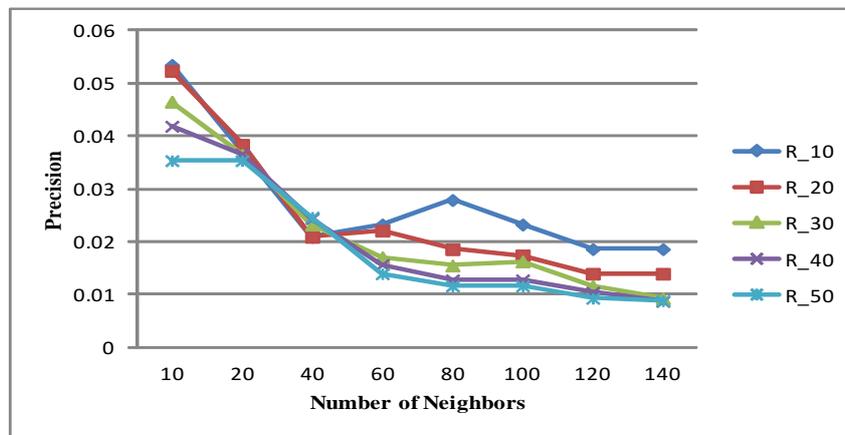
(a) Precision
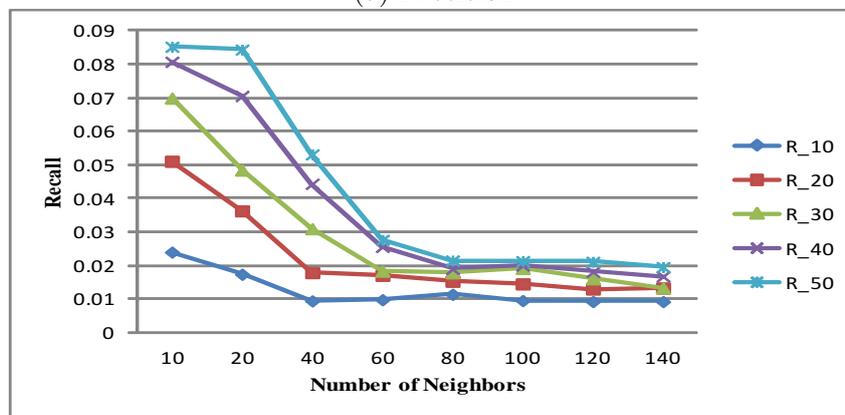
(b) Recall
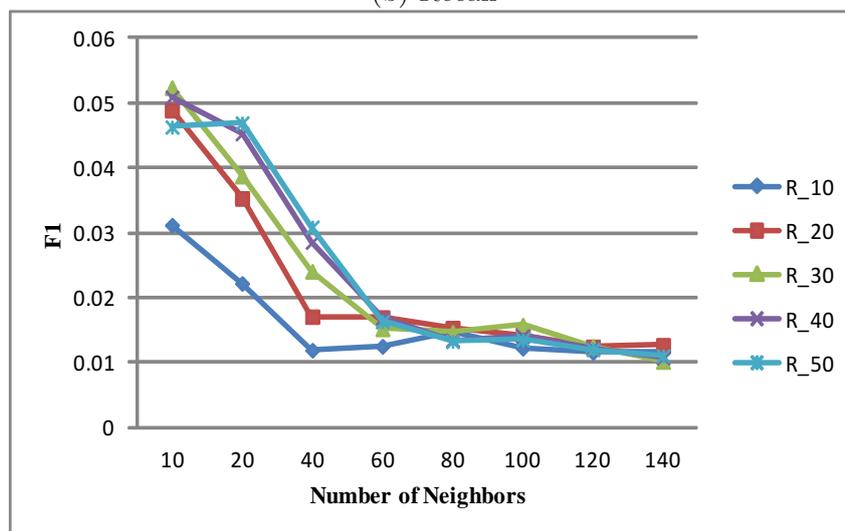
(c) F1

Figure 3: The effect of both the number of recommendations and the number of neighbors on the accuracy of proposed system
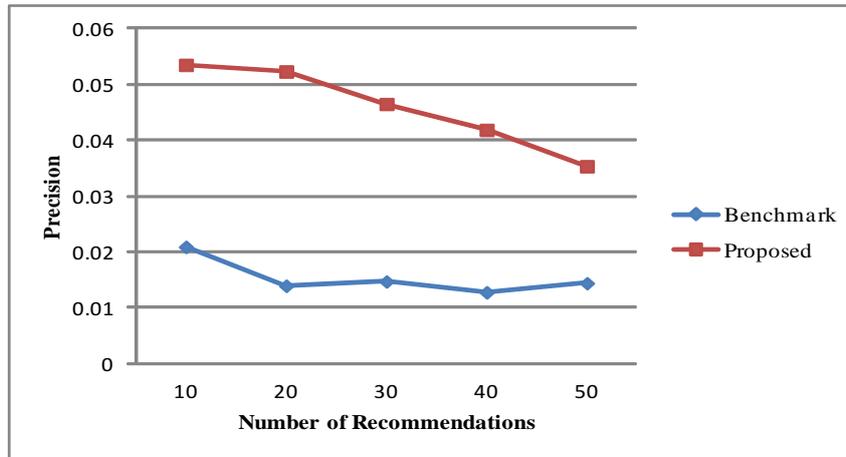
(a) Precision
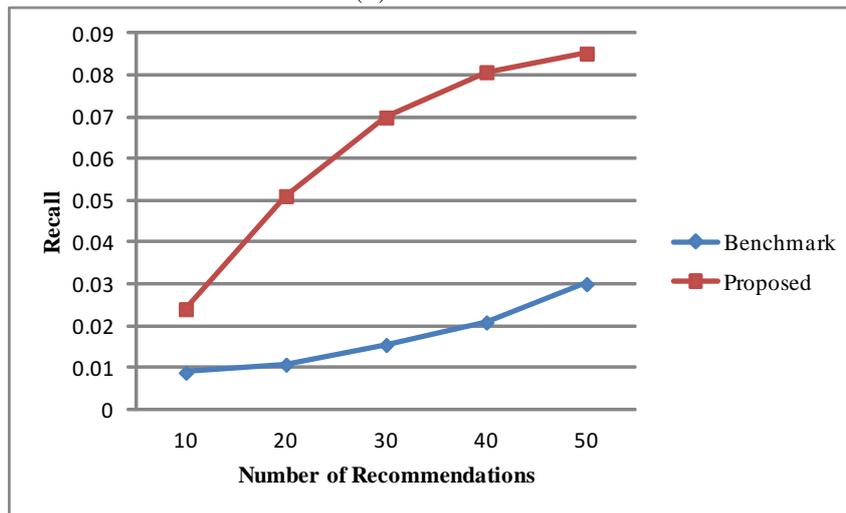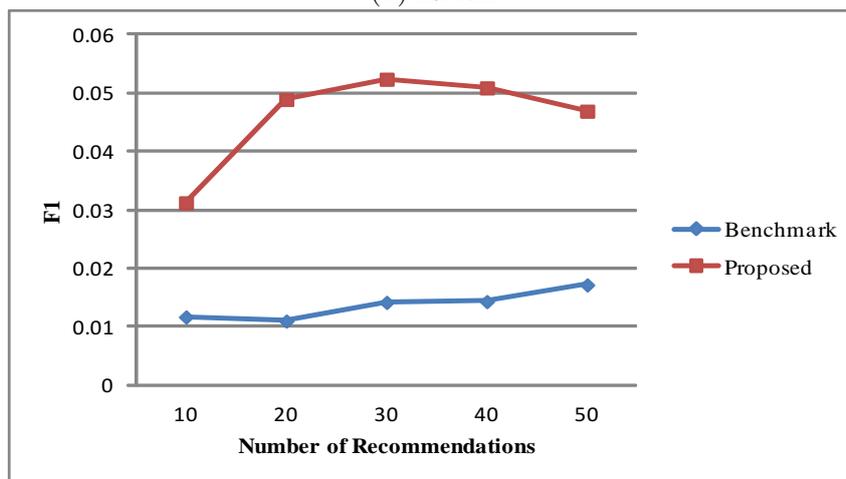
(b) Recall

(c) F1

Figure 4: A comparison between the benchmark system and the proposed system

(i.e., precision at $k = 120$ ($n = 10$) and 40 ($n = 20, 30, 40,$ and 50), recall at $k = 120$ ($n = 10$ and 20), 40 ($n = 30$), and 10 ($n = 40, 50$), and F1 at $k = 120$ ($n = 10, 20$), 40 ($n = 30$ and 50), and 60 ($n = 40$)) with those of our proposed system (i.e., precision at $k = 10$ ($n = 10, 20, 30, 40$) and 20 ($n = 50$), recall at $k = 10$ ($n = 10, 20, 30, 40,$ and 50), and F1 at $k = 10$ ($n = 10, 20, 30,$ and 40) and 20 ($n = 50$)). As shown in Fig. 4 (a), (b), and (c), our proposed system considerably outperformed the benchmark system in precision, recall, and F1, regardless of the number of recommendations. The results from our experiments proved that our approach to extending collaborative filtering technique to consider all pairs of similar items when computing the user similarity is effective on mitigating the sparsity problem, thereby enhancing the accuracy of recommendation systems.

5 Conclusions

The collaborative filtering technique has been suggested as one of the best methods for making recommendations and has proven to be useful in many applications, but that technique is not easy to use for recommendations in online shopping malls, because explicit rating information is rarely available. In addition, one of the problems of the technique, the sparsity problem, occurs due to the low level of customer input into the recommendation system in online shopping malls. Therefore, online shopping malls which have to survive high competition must resolve these two problems to be able to make effective recommendations which in turn improve their customer retention rate.

With an objective to provide online shopping malls with a way to make better recommendations this paper first explains how to derive implicit rating information from the transaction data that will replace the explicit rating information. It then defines a new user similarity function which computes a user similarity between two users by taking into account item similarity of all pairs of similar items, rated by these users.

In order to compare our proposed recommendation system with a traditional recommendation system, we implemented both systems and conducted several experiments. The results obtained from these experiments indicate that our approach considerably outperformed the traditional collaborative filtering approach in precision, recall, and F1. This study, however, leaves something to be desired. More reliable and interesting results could be obtained if we have used bigger datasets from more than one online shopping mall over a longer period of time.

Bibliography

[1] M. Pazzani, D. Billsus (1997); Learning and Revising User Profile: The Identification of Interesting Web Sites, *Machine Learning*, 27(3): 313-331.

[2] T. Zhang, R. Agarwal, H.C. Lucas (2011); The Value of IT-Enabled Retailer Learning: Personalized Product Recommendations and Customer Store Loyalty in Electronic Markets, *MIS Quarterly*, 35(4): 859-881.

[3] Y. Jing, H. Liu (2013); A Model for Collaborative Filtering Recommendation in E-Commerce Environment, *International Journal of Computers Communications and Control*, 8(4): 560-570.

[4] D.R. Liu, C.H. Lai, W.J. Lee (2009); A Hybrid of Sequential Rules and Collaborative Filtering for Product Recommendation, *Information Sciences*, 179(20): 3505-3519.

[5] M. Balabanovic, Y. Shoham (1998); Content-Based, Collaborative Recommendation, *Communications of the ACM*, 40(3): 66-72.

[6] K. Lang (1995); NewsWeeder: Learning to Filter Netnews, *Pro. of the 12th Int. Conference on Machine Learning*.

[7] U. Shardanand, P. Maes (1995); Social Information Filtering Algorithms for Automating "Word of Mouth", *Pro. of the SIGCHI Conference on Human Factors in Computing Systems*.

[8] G. Adomavicius, A. Tuzhilin (2005); 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(6): 734-749.

[9] D. Billsus, M.J. Pazzani (1998); Learning Collaborative Information Filters, *Pro. of the 15th Int. Conference on Machine Learning*.

[10] V. Formoso, D. Fernandez, F. Cacheda, V. Carneiro (2012); Using Profile Expansion Techniques to Alleviate the New User Problem, *Information Processing and Management*, 49(3): 659-672.

[11] H.N. Kim, A.T. Ji, I. Ha, G.S. Jo (2010); Collaborative Filtering based on Collaborative Tagging for Enhancing the Quality of Recommendation, *Electronic Commerce Research and Applications*, 9(1): 73-83.

[12] T.Q. Lee, Y. Park, Y.T. Park (2008); A Time-based Approach to Effective Recommender Systems Using Implicit Feedback, *Expert Systems with Applications*, 34(4): 3055-3062.

[13] Q. Shambour, J. Lu (2011); A Hybrid Trust-Enhanced Collaborative Filtering Recommendation Approach for Personalized Government-to-Business e-Services, *International Journal of Intelligent Systems*, 26(9): 814-843.

[14] C.C. Aggarwal, C. Procopiuc, P.S. Yu (2002); Finding Localized Associations in Market Basket Data. *IEEE Transactions on Knowledge and Data Engineering*, 14(1): 51-62.

[15] C.L. Huang, W.L. Huang (2009); Handling Sequential Pattern Decay: Developing a Two-Stage Collaborative Recommendation System, *Electronic Commerce Research and Applications*, 8(3): 117-129.

[16] Y. Wang, W. Dai, Y. Yuan (2008); Website Browsing Aid: A Navigation Graph-based Recommendation System, *Decision Support Systems*, 45(3): 387-400.

[17] D. Goldberg, D. Nichols, B.M. Oki, D. Terry (1992); Using Collaborative Filtering to Weave an Information Tapestry, *Communications of the ACM*, 35(12): 61-70.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl (1994); GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Pro. of the 1994 ACM Conference on Computer Supported Cooperative Work*.

[19] G. Adomavicius, Y. Kwon (2007); New Recommendation Techniques for Multicriteria Rating Systems, *IEEE Intelligent Systems*, 22(3): 48-55.

[20] I.S. Altingovde, O.N. Subakan, O. Ulusoy (2012); Cluster Searching Strategies for Collaborative Recommendation Systems, *Information Processing and Management*, 49(3): 688-697.

[21] K.W. Cheung, J.T. Kwok, M.H. Law, K.C. Tsui (2003); Mining Customer Product Ratings for Personalized Marketing, *Decision Support Systems*, 35(2): 231-243.

[22] K. Goldberg, T. Roeder, D. Gupta, C. Perkins (2001); Eigentaste: A Constant Time Collaborative Filtering Algorithm, *Information Retrieval*, 4(2): 133-151.

[23] Y. Koren (2010); Factor in the Neighbors: Scalable and Accurate Collaborative Filtering, *ACM Transactions on Knowledge Discovery from Data*, 4(1): 1-24.

[24] R. Salakhutdinov, N. Srebro (2010); Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm, arXiv:1002.2780v1, 1-9.

[25] D. Joaquin, I. Naohiro (1999); Memory-based Weighted-Majority Prediction. *Pro. of ACM SIGIR' 99 Workshop on Recommender Systems: Algorithms and Evaluation.*

[26] J. Lee, S. Lee, H. Kim (2011); An Probabilistic Approach to Semantic Collaborative Filtering Using World Knowledge, *Journal of Information Science*, 37(1): 49-66.

[27] J.M. Yang, K.F. Li, D.F. Zhang (2009); Recommendation based on Rational Inferences in Collaborative Filtering, *Knowledge-Based Systems*, 22(1): 105-114.

[28] Z. Liu, W. Qu, H. Li, C. Xie (2010); A Hybrid Collaborative Filtering Recommendation Mechanism for P2P Networks, *Future Generation Computer Systems*, 26(8): 1409-1417.