Expert Systems with Applications 28 (2005) 67–77

Expert Systems with Applications

www.elsevier.com/locate/eswa

A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce☆

Yu Li[a,b,*], Liu Lu[b], Li Xuefeng[b]

[a]*School of Information, Renmin University of China, Beijing 100872, China*
[b]*School of Economics and Management, BeiHang University, Beijing 100083, China*

Abstract

Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for products or services during a live interaction. These systems, especially collaborative filtering based on user, are achieving widespread success on the Web. The tremendous growth in the amount of available information and the kinds of commodity to Web sites in recent years poses some key challenges for recommender systems. One of these challenges is ability of recommender systems to be adaptive to environment where users have many completely different interests or items have completely different content (We called it as Multiple interests and Multiple-content problem). Unfortunately, the traditional collaborative filtering systems can not make accurate recommendation for the two cases because the predicted item for active user is not consist with the common interests of his neighbor users. To address this issue we have explored a hybrid collaborative filtering method, collaborative filtering based on item and user techniques, by combining collaborative filtering based on item and collaborative filtering based on user together. Collaborative filtering based on item and user analyze the user-item matrix to identify similarity of target item to other items, generate similar items of target item, and determine neighbor users of active user for target item according to similarity of other users to active user based on similar items of target item.

In this paper we firstly analyze limitation of collaborative filtering based on user and collaborative filtering based on item algorithms respectively and emphatically make explain why collaborative filtering based on user is not adaptive to Multiple-interests and Multiple-content recommendation. Based on analysis, we present collaborative filtering based on item and user for Multiple-interests and Multiple-content recommendation. Finally, we experimentally evaluate the results and compare them with collaborative filtering based on user and collaborative filtering based on item, respectively. The experiments suggest that collaborative filtering based on item and user provide better recommendation quality than collaborative filtering based on user and collaborative filtering based on item dramatically.

Keywords: Collaborative filtering; Recommender systems; Personalization; E-Commerce

1. Introduction

The amount of information in the world is increasing far more quickly than our ability to process it. All of us have known the feeling of being overwhelmed by the number of new books, journal articles, and conference proceedings coming out each year. Now it is time to create the technologies that can help us sift through all the available information to find what is most valuable to us.

☆ This work was supported by the National Science Foundation of China under grants No.70371004 and PhD program Foundation of Education Ministry of China contract number 2000000601.
* Corresponding author. Tel.: +8610 82324786.
E-mail address: buaayuli@hotmail.com (Y. Li).

0957-4174/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2004.08.013

One solution to this information overload problem is the use of recommender systems. Recommender systems are used by E-commerce sites to suggest products to their customers and to provide consumers with information to help them determine which products to purchase. The products can be recommended based on the top overall sellers on a site, on the demographics of the consumer, or on an analysis of the past buying behavior of the consumer as a prediction for future buying behavior. The forms of recommendation include suggesting products to the consumer, providing personalized product information, summarizing community opinion, and providing community critiques. Recommender systems enhance E-commerce sales in three ways: helping customers find products they wish to purchase; converting browsers into buyers; improving cross-sell by suggesting additional products for the customer to purchase; improving loyalty by creating a value-added relationship between the site and the customer (Schafer et al., 2001).

The underlying techniques used in today's recommendation systems fall into two distinct categories: content-based methods and collaborative methods. In general, a content-based system analyzes a set of documents rated by an individual user and uses the content of these documents, as well as the provided ratings, to infer a profile that can be used to recommend additional items of interest. However, collaborative methods recommend items based on aggregated user ratings of those items, i.e. these techniques do not depend on the availability of textual descriptions. Both approaches share the common goal of assisting in the user's search for items of interest, and thus attempt to address one of the key research problems of the information age: locating needles in a haystack that is growing exponentially.

In this paper we focus on collaborative filtering techniques. A variety of algorithms have previously been reported in the literature and their promising performance has been evaluated empirically (Breese, Heckerman, & Kadie, 1998; Hill, Stead, Rosenstein, & Furnas, 1995; Resnick, Iacovou, Suchak, Bergstrom, & Reidl, 1994; Sarwar, Karypis, Konstan, & Riedl, 2000; Shardanand & Maes, 1995). These results, and the continuous increase of people connected to the Internet, led to the development and employment of numerous collaborative filtering systems. Virtually all topics that could be of potential interest to users are covered by special-purpose recommendation systems: web pages, news stories, movies, music videos, books, CDs, restaurants, and many more. Some of the best-known representatives of these systems, such as *FireFly* (www.firefly.com) or *WiseWire* (www.wisewire.com) have turned into commercial enterprises. Furthermore, collaborative filtering techniques are becoming increasingly popular as part of online shopping sites. These sites incorporate recommendation systems that suggest products to users based on products that like-minded users have ordered before, or indicated as interesting. For example, users can find out which CD they should order from an online CD store if they provide information about their favorite artists, and several online bookstores (e.g. www.amazon.com) can associate their available titles with other titles that were ordered by like-minded people.

However, there remain important research questions in overcoming fundamental challenges for recommender systems based on collaborative filtering.

1.1. Related work

In this section we briefly present some of research literature related to collaborative filtering, recommender systems.

Tapestry (Goldberg, Nichols, Oki, & Terry, 1992) is one of the earliest implementations of collaborative filtering-based recommender systems. This system relied on the explicit opinions of people from a close-knit community, such as an office workgroup. However, recommender system for large communities cannot depend on each person knowing the others. Later, several ratings-based automated recommender systems were developed. The GroupLens research system (Konstan et al., 1997; Resnick et al., 1994) provides a pseudonymous collaborative filtering solution for Usenet news and movies. Ringo (Shardanand & Maes, 1998) and Video Recommender (Hill et al., 1995) are email and web-based systems that generate recommendations on music and movies respectively. A special issue of Communications of the ACM (Resnick & Varian, 1997) presents a number of different recommender systems. Other technologies have also been applied to recommender systems, including Bayesian networks, clustering, and Horting (Aggarwal, Wolf, Wu, & Yu, 1999; Breese et al., 1998).

Various approaches for recommender systems have been developed that utilize either demographic, content, or historical information (Balabanovic & Shoham, 1997; Basu, Hirsh, & Cohen, 1998; Hill et al., 1995; Konstan et al., 1997; Shardanand & Maes, 1995; Terveen, Hill, Amento, McDonald, & Creter, 1997). Collaborative filtering based on user (CF-U), is probably the most successful and widely used techniques for building recommender systems (Konstan et al., 1997; Resnick et al., 1994). For each user, collaborative filtering based on user use historical information to identify a neighborhood of people that in the past have exhibited similar behavior (e.g. accessed the same type of information, purchased a similar set of products, liked/disliked a similar set of movies) and then analyze this neighborhood to identify new pieces of information that will be liked by the user. So this method also be called neighbor-based collaborative filtering or nearest neighbor algorithms.

Unfortunately, user-based algorithms require computation that grows linearly with the number of users and items. With millions of users and items, user-based recommender systems suffer serious scalability problems. By virtue of scalability problems, an alternate approach, item-based collaborative filtering is to build recommendation models that are based on the items (Sarwar et al., 2000). In these approaches, the historical information is analyzed to identify relations between the items such that the purchase of an item (or a set of items) often leads to the purchase of another item (or a set of items) (Billsus & Pazzani, 1998; Breese et al., 1998; Kitts, Freed, & Vrieze, 2000; Wolf, Aggarwal, Wu, & Yu, 1999). These approaches, since they use the pre-computed model, can quickly recommend a set of items, and have been shown to produce recommendation results that in some cases are comparable to traditional, user-based CF recommender systems.

Schafer present a detailed taxonomy and examples of recommender systems used in E-commerce and how they can provide one-to-one personalization and at the same can capture customer loyalty (Schafer et al., 1999).Although these systems have been successful in the past, their widespread use has exposed some of their limitations such as the problems of sparsity in the data set, problems associated with high dimensionality and so on. Sparsity problem in recommender system has been addressed in (Good et al., 1999; Sarwar et al., 1998). The problems associated with high dimensionality in recommender systems have been discussed in (Billsus & Pazzani, 1998), and application of dimensionality reduction techniques to address these issues has been investigated in (Sarwar et al., 2000).

But very few work show that classical collaborative filtering is not adaptive to Multiple-interests recommendation. In fact, the quality of its recommendation is very poor when users in recommender systems have completely different interests. Unfortunately, this situation exists commonly. Once (Hofmann, 2001) tried to solve the problem using a probabilistic model from the model-based perspective, present the probabilistic model based collaborative filtering. But the method has the shortcoming of all model-based collaborative filtering. Firstly, the method presented by (Hofmann, 2001) cannot explain why classic collaborative filtering cannot adapt to multiple interests recommendation. Secondly, although the computing speed is far than user-based collaborative filtering, recommendation cannot vary with the rating database of recommendation system, which results that users can get on-line recommendation.

In this paper, we focus on solving this problem. Our work explores a new recommender algorithms, which is collaborative filtering based on users and items, are able to solve these problems.

1.2. Contributions

This paper has three primary research contributions:

1. Identification and analysis on limitation of collaborative filtering based on user for Multiple-interests and Multiple-content recommedation
2. Presentation of a hybrid collaborative filtering method, collaborative filtering based on item and user algorithm, to improve recommendation accuracy for Multiple-interests and Multiple-content recommedation
3. An experimental comparison of the quality of collaborative filtering based on item and user algorithm with traditional collaborative filtering based on user algorithm and collaborative filtering based on item algorithm.

1.3. Organization

The rest of the paper is organized as follows. The next section provides a brief background on collaborative filtering based on user algorithms and collaborative filtering based on item algorithms. We first formally describe the two kinds of collaborative filtering algorithms. In Section 3, we analyze questions of collaborative filtering based on user for Multiple-interests and Multiple-content recommendation. In Section 4, we present a hybrid collaborative filtering method, collaborative filtering based on item and user algorithm. Section 5 describes our experimental work. It provides the details of data sets, evaluation metrics, procedure and results of different experiments, as well as the discussion of the results. Section 6 provides some concluding remarks and directions for future research.

2. Collaborative filtering (CF)

Recommender systems apply data analysis techniques to the problem of helping users find the items they would like to purchase at E-Commerce sites by producing a predicted likeliness score or a list of top-N recommended items for a given user. Collaborative filtering is most important personalized recommendation method widely in recommender systems. Currently, there are two kinds of collaborative filtering algorithms.

2.1. Collaborative filtering based on user (CF-U)

Collaborative filtering based on user (Resnick et al., 1994; Sarwar et al., 2000; Shardanand and Maes, 1995) is the most successful recommending technique to date, and is extensively used in many commercial recommender systems. These schemes rely on the fact that each person belongs to a larger group of similarly behaving individuals. Consequently, items (i.e. products) frequently purchased by the various members of the group can be used to form the basis of the recommended items.

Let R be an $n \times m$ user-item matrix containing historical purchasing information of $n$customers on m items. In this matrix, $R_{i,j}$ is one if the ith customer has purchased the jth item, and zero otherwise. Let I_R be the set of items that have already been purchased by the customer for which we want to compute the top-N recommendations. We will refer to this customer as the active customer and in order to simplify the presentation we will assume that the active customer does not belong to the n customers stored in matrix R. Recommender systems based on CF-U compute the top-N recommended items for that user as follows.

First they identify the k most similar users in the database. This is often done by modeling users and items with the vector-space model, which is widely used for information retrieval (Breese et al., 1998, Sal89, Sarwar et al., 2000). In this model each of the n users as well as the active user is treated as a vector in the m-dimensional item space, and the similarity of active user to existing users is measured by computing the cosine between these vectors or correlation.

Once this set of the k most similar users have been discovered, their corresponding rows in R are aggregated to identify the set C of items purchased by the group as well as their frequency. Using this set, CF-U techniques then recommend the N most frequent items in C that are not already in I_N (i.e. the active user has not already purchased). Note that the frequency of the items in the set C can be computed either by counting the actual occurrence frequency or by first normalizing each row of R to be of the same length. This latter normalization gives less emphasis to items purchased by customers that are frequent buyers and leads to somewhat better results.

Despite the popularity of CF-U, it has a number of limitations related to scalability and real-time performance. The computational complexity of these methods grows linearly with the number of users, which can grow to be several millions in typical commercial applications. Furthermore, recommender systems based on CF-U are hard to provide the explanation for the recommendation.

Because of the two limitations above, an alternate approach, collaborative filtering based on item algorithm is presented to build recommendation models that are based on the items.

2.2. Collaborative filtering based on item (CF-I)

To address the scalability concerns of CF-U algorithms and provide better explaining for recommendation to users, collaborative filtering based on item (CF-I) techniques have been developed (Billsus and Pazzani, 1998; Breese et al., 1998; Wolf et al., 1999, Kar00, Sarwar et al., 2000). These approaches analyze the user-item matrix to identify relations between the different items, and then use these relations to compute the list of top-N recommendations. The key motivation behind these schemes is that a user will be more likely purchase items that are similar or related to the items that he/she has already purchased. Since these schemes do not need to identify the neighborhood when a recommendation is requested, they lead to much faster recommendation engines. A number of different schemes have been proposed to compute the relations between the different items based on either probabilistic approaches or more traditional item-to-item correlations.

During the model building phase, firstly, for each item j, the k most similar items $\{j_1, j_2,\ldots,j_k\}$ are computed, and their corresponding similarities $\{w_{j1}, w_{j2},\ldots,w_{jk}\}$ are recorded. Then prediction value is computed by summing the rating of active user for the k most items $\{j_1, j_2,\ldots,j_k\}$ with weight $\{w_{j1}, w_{j2},\ldots,w_{jk}\}$.Finally, the items that are not ratedare sorted in non-increasing order with respect to prediction value, and the first N items are selected as the top-N recommended set.

2.3. Limitations of CF-U and CF-I

Although CF-I algorithm overcomes the scalability of CF-U, and can provide better explaining for recommendation result, it cannot provide novelty and personalized recommendation for user. Moreover, there are experiments showing that CF-U provides better recommendation (Mild & Natter, 2001). In almost case, accuracy of CF-I algorithm is poorer than CF-U algorithm.

Except for scalability problem discussed by many literatures, CF-U has another limitation, which provides much poor recommendation if users have many different interests or items have completely different content, which we called as Multiple-interests and Multiple-content problem. Unfortunately, the cases exist commonly. The limitation results that users do not trust on recommender system. The fact has shown that users usually do not seek for help by recommender system when they purchase costly commodity.

In this paper, we will focus on Multiple-interests and Multiple-content problem and try to resolve them by combining CF-I and CF-U.

3. Problem analysis

In this section, firstly, we mainly analyze Multiple-interests and Multiple-content problem of CF-U algorithm in detail. Then new method is presented.

3.1. Problem statements

According to CF-U, prediction for target item is determined by preference of user for rated item (also called as history items). But it is common that one user have many different interests (Multiple-interests) and items have completely content (Multiple-content), but history items or rated item are on one interests or content, and predicted item is on another interests or content. The fact makes it possible that predicted item has no related to history items. For example, a user may be interested in items related to ‘Football’ and ‘English’, but only rated Football items. Now if we predict English item, then interest preference of the user for Football items will only be used, accuracy of prediction result is doubtful.

Let’s see an example as following.

As shown in Table 1, in user/item data matrix, there are seven users and six items. We supposed that, content of item I1 (English), I3 (English) and I6 (English) is on English, that is to say the three items are similar on content, but are three different items. In the same way, content of item I2 (Football), I4 (Football) and I5 (Football) is on Football, which is different from content of item I1, I3 and I6, is another content or another interest of user.

Supposed we will predict rating of user U7 for item I6, $R_{76}=?$ (Here user U7 is active user, item I6 is predicted item.). We supposed that each user has three neighbor users.

If prediction is done according to CF-U algorithm, it is obvious that users U4, U5, and U6 will be neighbors of U7 because of their similar rating behavior to user U7.It is easy to get the prediction value, $R_{76}=1$. But we find that the reason why users U4, U5, and U6 will be neighbors of active user U7 is mainly that all of them are interested in football

Table 1
An example of user/item data matrix

User	Item					
	I1 (English)	**I2** (Football)	**I3** (English)	I4(**Football)**	**I5** (Football)	**I6** (English)
U1	3	1	2	3	5	5
U2	3	1	2	3	5	5
U3	3	1	2	3	5	5
U4	1	5	3	3	1	1
U5	2	5	2	3	2	1
U6	3	5	1	3	2	1
U7	3	5	2	4	2	?

sport, which is their common interest. That is to say that we used interest preference of users on items related to football (I2, I4, I5) to predict interest preference on item related to English (I6), but Football and English are not related and dependent, so prediction is not accurate, and recommendation based on prediction is doubtful.

In order to further illustrate our view, let's see an extreme example. For Table 1, if there were no ratings of users U4, U5 and U6 for items I1 and I3 (this kind of case often occurs because users often only rate a very few items and most items do not rated for a user), then we get the Table 2. Likewise, here we predict rating of user U7 for item I6, $R_{76}=?$

According to CF-U algorithm, users U4, U5, U6 will be neighbors of user U7, and prediction value of rating of U7 for I6 is completely determined by their preference information on items related to football (I2, I4, I5) because users U4, U5, U6 and active user U7 have complete same rating behavior. But we must observe that common interest in items related to football (I2, I4, I5) of users U4, U5, U6 and U7 is no related to predicted item (I6) because item on English has completely different content from items on football. So, prediction is not accurate.

3.2. Our solution

For above example, if neighbors of active user U7 are determined according to interest preference of all users for items related to English (that is to say, similarity of user U7 to other users are computed according to ratings of all users for I1 and I3, not for all items), then we will find that users U1, U2. U3 will be neighbors of active user U7. So predicted value, $p_{76}=5$ will more be trustful because common interest preference of active user U7 and its neighbors U1, U2. U3 for English is similar, and rating of neighbors U1, U2. U3 for item I6 can be used to predict rating of active user U7 for item I6.

According to above analysis, we can get a conclusion that active user should has common interest in predicted item with its neighbor users. So, computation of similarity of active user to other users should be based on items related to predicted item, not on all items, more being not based on items not related to predicted item. It means that, for different predicted item, neighbors of the same active user are different. It can guarantee that predicted item and items used to predict for predicted item are similar on content, which can lead to improvement for CF-U. According the solution, we present a hybrid collaborative filtering method, collaborative filtering based on item and user algorithm.

4. Hybrid collaborative filtering method

According to above analysis, we present a hybrid collaborative filtering method, collaborative filtering based on item and user (CF-IU), by combining collaborative filtering based on item (CF-I) and collaborative filtering based on user (CF-U). The new method is adaptive to Multiple-interest and Multiple-content recommendation.

First, we denoted,

(1) $R_{i,j}$—Rating of user *i* for item *j*, which reflect the interest preference of user *i* for item *j*;

Table 2
An extreme example

User	Item					
	I1 (English)	**I2** (Football)	**I3** (English)	I4(**Football)**	**I5** (Football)	**I6** (English)
U1	3	1	2	3	5	5
U2	3	1	2	3	5	5
U3	3	1	2	3	5	5
U4	–	5	–	3	1	1
U5	–	5	–	3	2	1
U6	–	5	–	3	2	1
U7	3	5	2	3	2	?

(2) $I_{R,i}=\{i|R_{i,j}$ is not null, $j\in 1,2,\ldots,n\}$, denoted as item sets which user $i$had rated;
(3) $I_{P,i}=\{i|R_{i,j}$ is null, $j\in 1,2,\ldots,n\}$, denoted as item sets which user $i$had not rated;

Inputs and output of the algorithm are as below:

(I) Inputs:
1. Rating of m users for n items (allowing null rating);
2. Item correlation threshold—s, user correlation threshold—w, rating threshold—r;
3. *similar_items_number*—the maximal number of similar items for predicted item;
4. *neighbor_number*—the maximal number of neighbor users for active user;
5. *recom_items_number*—the maximal number of recommended items;

Users input information can be presented by user/item rating matrix, see Table 3.

(II) Outputs:
The most L interesting items in item sets $I_{p,a}$ for User a (named the predicted user a as *Active User*) are recommended to active user.

(III) Computing procedure:
1. For target item $j\in I$, Compute similarity between item $j$and other items $q(q\in 1,2,\ldots,j-1,j+1,n)$—$Sim(j,q)$;Similarity may be measured by *Pearson correlation coefficient*, *cosine* and so on;
2. According to $Sim(j,q)$ $(q\in 1,2,\ldots,j-1,j+1,n)$, determine similar item sets to target item $j-SI_j$; Two methods are used to determine SI_j:
   1. Thresholding: by setting an absolute item similarity threshold s, to determine similar items SI_j to target item with absolute similarity $Sim(j,q)$ greater than s.
   2. Best-itwms: picking up the maximal *similar_items_number* similarity from all $Sim(j,q)$, $(k\in 1,2,\ldots,j-1,j+1,n)$, and corresponding *similar_items_numbe* items form SI_j;
3. Compute similarity $w_j(a,i)$ of active user a to other user $i(i\neq a)$ based on item sets SI_j;
   1. Determine common rated item sets $CSI_j(a,i)$ of user a and user i;
   2. Compute similarity $w_j(a,i)$ of active user a to other user $i(i\neq a)$ based on $CSI_j(a,i)$ measured by *Pearson* correlation coefficient, *cosine* and other measurement.
4. According to similarity $w_j(a,i)$, determine neighbor users $Neighbor_{a,j}$ of active user $a$for target item j;Two methods are used to determine $Neighbor_{a,j}$:
   1. Thresholding: by setting an absolute user similarity threshold w, to determine neighbors $Neighbor_{a,j}$ of active user with absolute similarity $w_j(a,i)$ greater than w.
   2. Best-neighbors: picking up the maximal *neighbor_number* similarity from all $w_j(a,i)$ $(i\in 1,2,\ldots,a-1, a+1,\ldots,n)$, and corresponding *neighbor_number* users form $Neighbor_{a,j}$.
5. By performing a weighted average of deviations from mean of neighbors $Neighbor_{a,j}$, compute predicted value $P_{a,j}$ of rating of user a for item j,

$$P_{a,j}=\overline{R_a}+k\sum_{i\in \mathrm{Neighnor}_{a,j}} w_j(a,i)(R_{i,j}-\overline{R_i}),$$

$$\frac{1}{k}=\sum_{i\in \mathrm{Neighbor}_{a,j}} w_j(a,i)$$

6. Generate recommendation to active user a by selecting the most interesting items of user a according to $p_{a,j}$ $(j\in I_{P,a})$.Two methods are used to determine recommendation:
   1. Threshold: setting an rating threshold r, all prediction with value $p_{a,j}(j\in I_{P,a})$ greater than $r$are select to form recommendation;
   2. Best-L-rating: selecting the maximal *recom_ items_ number* prediction value from all $p_{a,j}(j\in I_{P,a})$ to form recommendation.

Table 3
Typical user/item rating matrix

User	Item					
	I1	**I2**	…	**Ij**	…	**In**
U1	R_{11}	R_{12}	…	R_{1j}	…	R_{1n}
U2	R_{21}	R_{22}	…	R_{2j}	…	R_{2n}
…	…	…	…	…	…	…
Ui	R_{i1}	R_{i2}	…	$\mathbf{R_{ij}=?}$	…	R_{in}
…	…	…	…	…	…	…
Um	R_{m1}	R_{m2}	…	R_{mj}	…	R_{mn}

5. Experimental evaluation

5.1. Datasets

We ran experiments using data from *EachMovie* collaborative filtering service. The *EachMovie* service was part of a research project at the Systems Research Center of Digital Equipment Corporation. The service was available for a period of 18 months and was shut down in September 1997. During that time the database grew to a fairly large size, containing ratings from 72,916 users on 1628 movies. User ratings were recorded on a numeric six-point scale (0.0, 0.2, 0.4, 0.6, 0.8, 1.0). The data set is publicly available and can be obtained from Digital Equipment Corporation (McJones, 1997).

Although the data from 72,916 users is available, we restrict the analysis to the first 450 users in the database. These 450 users provided ratings for 300 different movies. We restricted the number of users and items considered, because we are only interested in the performance of the algorithm under conditions where the number of users and items is low. This is a situation, through which every collaborative filtering service has to go in its startup-phase, and in many domains we cannot expect to have many users rating for many items. We also believe that the deficiencies of CF-IU algorithm will be more noticeable under these conditions, because it is less likely to find users with considerable overlap of rated items.

5.2. Evaluation metrics

Recommender systems research has used several types of measures for evaluating the quality of a recommender system. They can be mainly categorized into two classes:

Statistical accuracy metrics evaluate the accuracy of a system by comparing the numerical recommendation scores against the actual user ratings for the user-item pairs in the test dataset. *Mean Absolute Error* (MAE) between ratings and predictions is a widely used metric. MAE is a measure of the deviation of recommendations from their true user-specified values. For each ratings-prediction pair $\langle p_i, q_i \rangle$, this metric treats the absolute error between them i.e., $|p_i - q_i|$ equally. The MAE is computed by first summing these absolute errors of the Ncorresponding ratings—prediction pairs, and then computing the average. Formally,

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N}$$

The lower the MAE is, the more accurately the recommendation engine predicts user ratings. *Root Mean Squared Error* (RMSE), and *Correlation* are also used as statistical accuracy metric.

Decision support accuracy metrics evaluate how effective a prediction engine is at helping a user select high quality items from the set of all items. These metrics assume the prediction process as a binary operation—either items are predicted (good) or not (bad). With this observation, whether a item has a prediction score of 1.5 or 2.5 on a five-point scale is irrelevant if the user only chooses to consider predictions of 4 or higher. The most commonly used decision support accuracy metrics are *reversal rate*, *weighted errors* and *ROC sensitivity* (Sarwar et al., 1998).

We used MAE as the choice of evaluation metric to report prediction experiments because it is most commonly used and easiest to interpret directly. In the paper (Sarwar et al., 1998), the experiments have shown that MAE and ROC provide the same ordering of different experimental schemes in terms of prediction quality.

5.3. Experimental procedure

Our experiments are divided into three parts.

The first part compared CF-IU algorithm with CF-U algorithm. Because accuracy of the two algorithms have a close relation to number of user, we decide to test the sensitivity CF-IU and CF-U to number of user. So, we fixed number of items in training and testing set as 250 (Item is movie in *EachMovie* dataset), and varied the number of users from 50 to 450. In this way, we test five small datasets, called as U50M250, U150M250, U250M250, U350M250 and U450M250. For each experimental dataset, training sets include 200 movies and testing sets include 50 movie. In each experiment, we varied user correlation threshold (UCT) from 0 to 0.9 with step 0.1 for CF-U algorithm, and also varied UCT and movie correlation threshold (MCT) from 0 to 0.9 with step 0.1 for CF-IU algorithm. In CF-IU experiment, for each fixed UCT, we compute the average MAE of prediction for all varied MCT, and use it to compare with MAE of CF-U for corresponding UCT.

In the second part, we compared CF-IU algorithm with CF-I algorithm. Because quality of the two algorithms is more dependent on the number of items, we decided to test the sensitivity of CF-I and CF-IU to number of movies. For the reason, we fixed number of users in training and testing set as 250, and varied the number of movies from 100 to 300. So we tested five datasets, called as U250M100, U250M150, U250M200, U250M250 and U250M300. Each experiment dataset includes testing sets with 50 movies and training sets with different number of movies varied from 50 to 250. In each experiment, we varied MCT from 0 to 0.9 with step 0.1 for CF-I algorithm, and also varied UCT and MCT from 0 to 0.9 with step 0.1 for CF-IU. In CF-IU

Table 4
Experimental datasets

Data sets		Train count	Test count	All count	Train data density (%)	Test data density (%)	Users	Train movies	Test movies
Group (I)	U50M250	512	74	586	5.120	2.960	50	200	50
	U150M250	1130	173	1303	3.767	2.307	150	200	50
	U250M250	1598	270	1868	3.196	2.160	250	200	50
	U350M250	2166	389	2555	3.094	2.223	350	200	50
	U450M250	2772	585	3357	3.080	2.600	450	200	50
Group (II)	U250M100	846	297	1143	6.768	2.376	250	50	50
	U250M150	1143	209	1352	4.572	1.672	250	100	50
	U250M200	1352	246	1598	3.605	1.968	250	150	50
	U250M250	1598	270	1868	3.196	2.160	250	200	50
	U250M300	1868	354	2222	2.989	2.832	250	250	50

experiment, for each fixed MCT, we compute the average MAE of prediction for all varied UCT, and use it to compare with MAE of CF-I for corresponding MCT.

Finally, we analyze the performance of CF-IU algorithm for all above varied datasets (Table 4).

5.4. Experimental results

In this section we present our experimental results of applying collaborative filtering based on item and user techniques to generate prediction.

Our results are mainly divided into two parts, including quality results and performance results. In order to assess the quality of recommendations, we compared CF-IU with CF-I and CF-U for different UCT and MCT.

Based on the quality evaluation, we analyze the performance of CF-IU by determining the sensitivity to datasets. We test CF-IU in different datasets by varying the number of users for fixed the number of movies and by varying the number of movies for fixed the number of users.

5.4.1. Comparing CF-IU with CF-U

To compare quality of CF-IU and CF-U, we had made experiments on five datasets, and each dataset has identical movies (training set including 200 movies; testing set including 50 movies) and different the number of users, varied from 50 to 450. We denoted the five datasets as U50M250, U150M250, U250M250, U350M250 and U450M250, respectively. Because of similar results for some datasets, here we only show comparing results for three datasets, they are U50M250, U250M250, and U450M250.

The results are shown in Figs. 1–3. In the figures, *Y*-coordinate denotes mean average error (MAE), *X*-coordinate denotes different user correlation threshold (UCT).

It is shown in Figs. 1–3 that CF-IU is better than CF-U on quality for different UCT in three datasets experiments on the whole. And we find that its advantage is not very obvious when UCT is lower, but with UCT increased, CF-IU has higher recommending quality over CF-U.

Comparing Figs. 1–3 with each other, we can see that when user number in datasets is low, quality of CF-IU is not steady, and for some UCT, error of CF-IU is higher than CF-U. The finding falls in with our expectation because CF-IU requires more users in recommender systems than CF-U. But when users in datasets are more, CF-U has a higher error than CF-IU for all varied UCT. The results are enough to show that CF-IU has higher prediction quality than CF-U because there are often hundreds of users in recommender systems.

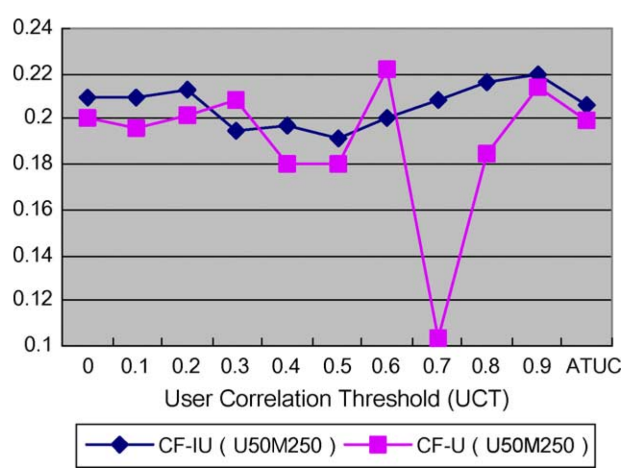

Fig. 1. Impact of UCT on recommending quality for CF-IU and CF-U (*Data set U50M250*).

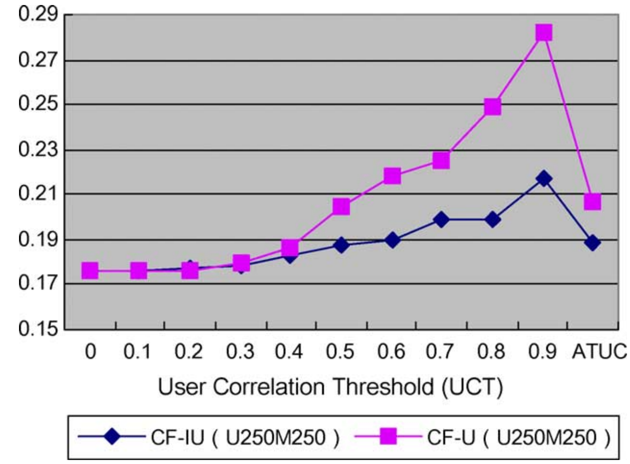

Fig. 2. Impact of UCT on recommending quality for CF-IU and CF-U (*Data set U250M250*).

5.4.2. Comparing CF-IU with CF-I

The second experiment is design to compare recommending quality of CF-IU and CF-I. For this task, we have also made experiments on the five datasets, and each dataset has identical users (250 users) and different number of movies varied from 100 to 300. We respectively denoted the five datasets as U250M100, U250M150, U250M200, U250M250 and U250M300. Because of similar results for

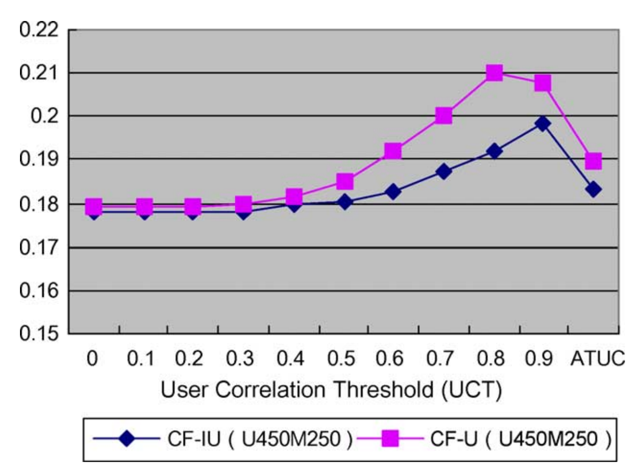

Fig. 3. Impact of UCT on recommending quality for CF-IU and CF-U (*Data set U450M250*).

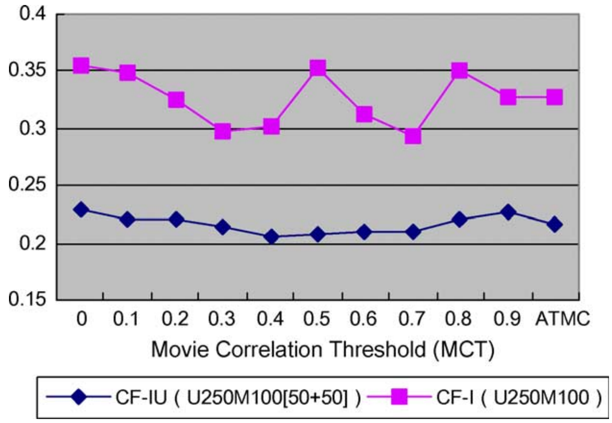

Fig. 4. Impact of MCT on recommending quality for CF-IU and CF-I (*Data set U250M100*).

some datasets, we only showed comparing results for three datasets (U250M100, U250M200 U250M300).

The results are shown in Figs. 4–6. In figures, *Y*-coordinate denotes mean average error (MAE), *X*-coordinate denotes different movie correlation threshold (MCT).

As we can see from Figs. 4–6, generally, CF-IU is better than CF-I on recommending quality for different MCT in three datasets experiments. At the same time, results presented in Figs. 4–6 show that CF-IU has a more steady quality for different MCT, and impact of MCT on the recommending quality for CF-I is greater than for CF-IU. Out of our expectation, in experiment on datasets U250M200, CF-IU has a higher error than CF-I when MCT is 0.8 and 0.9. We guess that different sparsity degree in same datasets accounts for the finding. But the exception does not affect making conclusion that CF-IU is better than CF-I on the whole.

For further verifying our results, in all datasets experiments, we compute average MAE of CF-IU, CF-I and CF-U for different UCT and MCT, and results are shown in Table 5 and Fig. 7. As we can clearly see from Fig. 7, the average MAE of CF-IU is the lowest for different

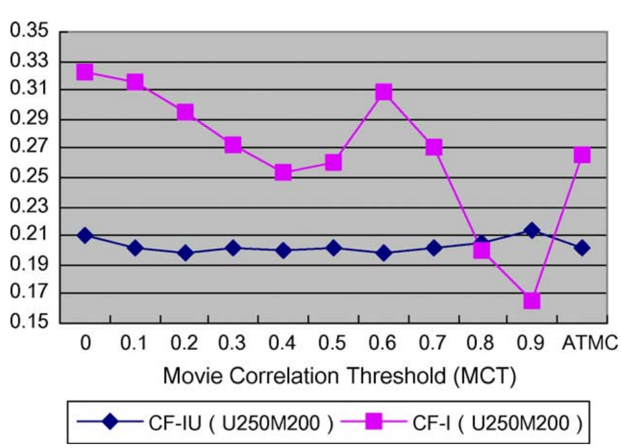

Fig. 5. Impact of MCT on recommending quality for CF-IU and CF-I (*Data set: U250M200*).

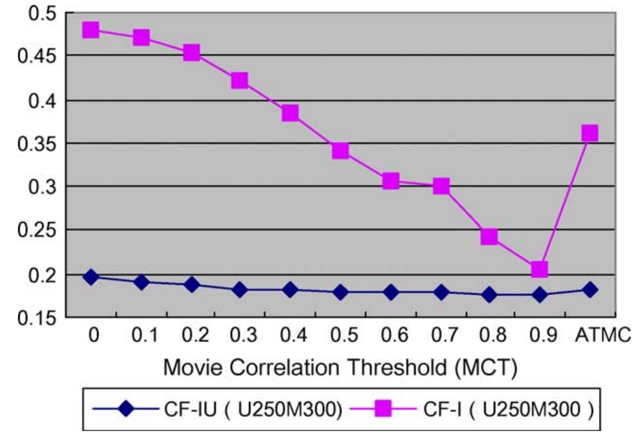

Fig. 6. Impact of MCT on recommending quality for CF-IU and CF-I (*Data set: U250M300*).

datasets. And the advantage of CF-IU over CF-I is more obvious than advantage of CF-IU over CF-U.

5.4.3. Performance of CF-IU

In this section, we will show how recommending quality of CF-IU changes with increasing or decreasing of users and movie, and learn the sensitivity of CF-IU to datasets. The experimental results are presented in Figs. 8–11. It is shown in these figures that, bigger the datasets is, higher the quality of CF-IU. But the result is not supported for a few datasets. We think that different sparsity of datasets is responsible for it.

5.5. Discussion

From experimental evaluation of CF-IU scheme we have got some important observations.

First, CF-IU provides better quality of predictions than CF-I scheme. The improvement in quality is consistent over different MCT and training/test ratio, and the improvement is very significantly large when dataset has more users. The second observation is CF-IU also provides better quality of predictions than CF-U. The improvement in quality is almost consistent over different UCT and training/test ratio.

Table 5
Average MAE of CF-U, CF-I, CF-IU algorithm for different UCT and MCT in different experimental datasets

Data sets	CF-U	CF-I	CF-IU
U50M250	0.199	0.272	0.2062
U150M250	0.234	0.3042	0.1937
U250M250	0.2074	0.3266	0.1882
U350M250	0.1939	0.3284	0.1839
U450M250	0.1894	0.3365	0.1832
U250M100	0.2258	0.3264	0.2162
U250M150	0.1947	0.3191	0.1889
U250M200	0.2011	0.2663	0.2023
U250M250	0.2074	0.3266	0.1882
U250M300	0.1958	0.3607	0.1824

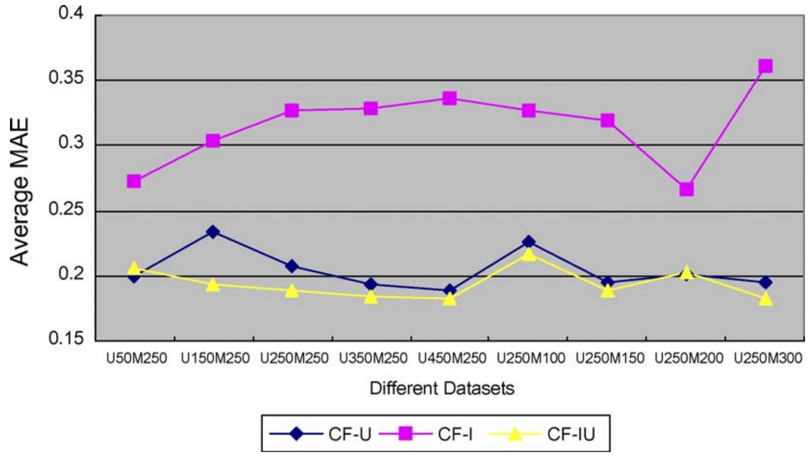

Fig. 7. Comparing on MAE of CF-IU, CF-I and CF-U algorithm for nine different datasets.

However, the improvement is not very obvious, especially when the dataset is small.

One important point is need to state, although the above experiments have shown some advantage of CF-IU over CF-U and CF-I, we believe that the advantage is greater if content of items are completely different in user/item matrix. Because all items of *EachMovie* datasets are movies and content of items are simplex, CF-IU cannot exhibit its merit that is dealing with Multiple-interesting and Multiple-content recommendation. Except for movie, supposed items of datasets also includes other commodity, such as book, CD, etc. and users are interested in many items that have completely content, then CF-IU be more accurate than CF-U. For this case, CF-IU has a greater advantage because the algorithm is able to filter dissimilar item to target item and to engender neighbor users of active user based on similar items to target item, which guarantee that target item is consist with the common interest of neighbor users, but CF-U cannot.

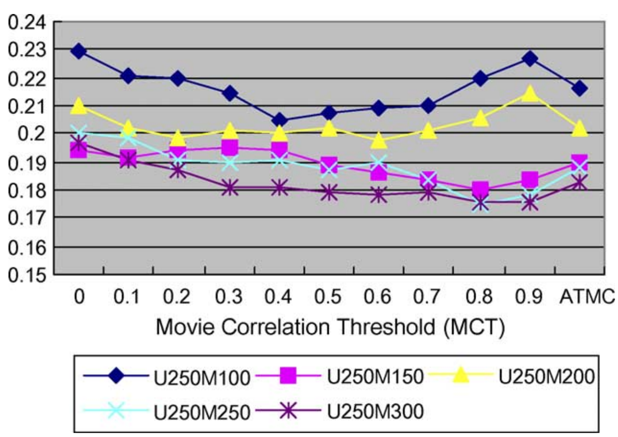

Fig. 8. Sensitivity of CF-IU algorithm to movies numbers and MCT.

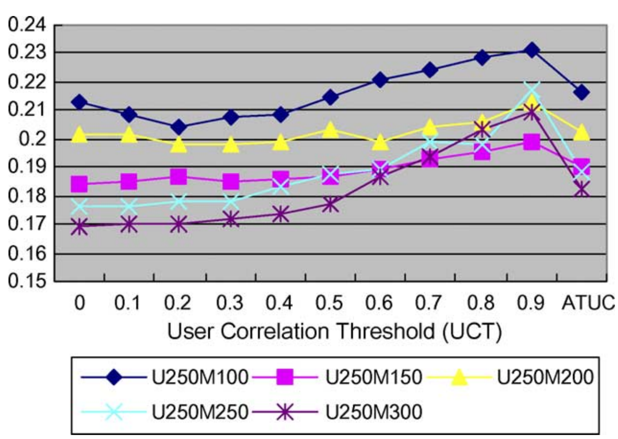

Fig. 9. Sensitivity of CF-IU algorithm to movies numbers and UCT.

6. Summary and further research

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on

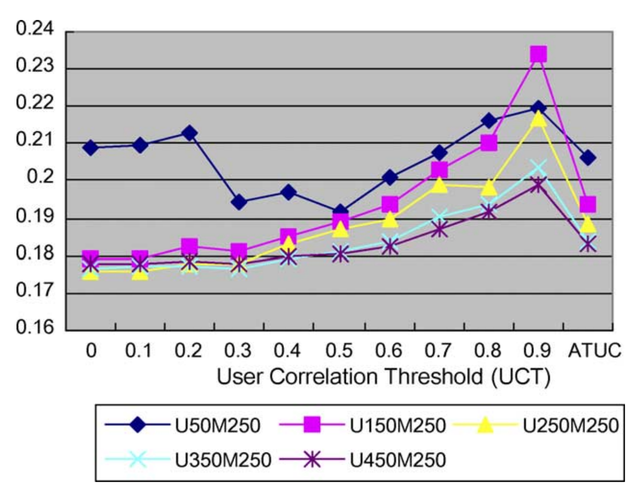

Fig. 10. Sensitivity of CF-IU algorithm to users numbers and UCT.

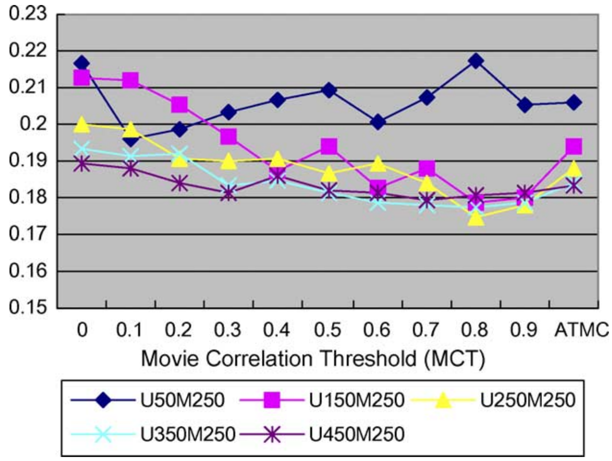

Fig. 11. Sensitivity of CF-IU algorithm to users numbers and MCT.

the Web. Collaborative filtering is the most important method widely used in recommender systems.

In this paper we presented and experimentally evaluated a new algorithm, collaborative filtering based on item and user. Our results showed that CF-UI algorithm hold the promise of allowing CF-based algorithms to be adaptive to data sets in which users have many different interests or items have completely content.

Our future study will be made from two directions. One is to apply collaborative filtering based on item and user to the application in E-Commerce. In addition, the research on how to give an explanation for recommendation is imperative for improving the confidence of recommender systems.

Acknowledgements

Funding for this research was supported by the National Science Foundation of China under grants No.70371004 and PhD program Foundation of Education Ministry of China contract number 2000000601. We would like to thank the System Research Center of Digital Equipment Corporation for making the *Each-Movie* database available for research. We also like to thank anonymous reviewers for their valuable comments.

References

Aggarwal, C. C., Wolf, J. L., Wu, K.-L., & Yu, P. S. (1999). Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. *Knowledge Discovery and Data Mining* , 201–212.

Balabanovic, M., & Shoham, Y. (1997). FAB: Content-based collaborative recommendation. *Communications of the ACM*, *40*(3).

Basu, C., Hirsh, H., & Cohen, W. (1998). *Recommendation as classification: Using social and content-based information in recommendation Recommender system workshop '98* pp. 11–15.

Billsus, D., & Pazzani, M. (1998). *Learning collaborative information filters Proceedings of the international conference on machine learning (Madison, WI)*. Los Altos: Morgan Kaufmann Publishers.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). *Empirical analysis of predictive algorithms for collaborative filtering Proceedings of the 14th conference on Uncertainly in Artificial Intelligence* pp. 43 C52.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information apestry. *Communications of the ACM*, *35*, 12.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). *Combining collaborative filtering with personal agents for better recommendations Proceedings of the sixteenth national conference on artificial intelligence (AAAI-99)* pp. 439–446.

Hofmann, T. (2001). *Learning what people (don't) want Proceedings of the European conference on machine learning*.

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). *Recommending and evaluating choices in a virtual community of use CHI '95: Conference proceedings on human factors in computing systems, Denver, CO* pp. 194–201.

Kitts, B., Freed, D., & Vrieze, M. (2000). *Cross-sell: A fast promotion-tunable customer-item recommendation method based on conditional independent probabilities Proceedings of ACM SIGKDD international conference* pp. 437–446.

Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., & Riedl, J. (1997). GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, *40*(3), 77–87.

McJones, P. (1997). *EachMovie collaborative filtering data set*. DEC Systems Research Center. http://www.research.digital.com/SRC/eachmovie/.

Mild, A., & Natter, M. A critical view on recommendation systems. Working paper series. Vienna University of Economics and Business Administration, Austria. http://www.wu-wien.ac.at/am.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *GroupLens: An open architecture for collaborative filtering of netnews Proceedings of the conference on computer supported cooperative work, Chapel Hill, NC* pp. 175–186.

Resnick, & Varian (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56 C58.

Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). *Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system Proceedings of CSCW '98, Seattle, WA*.

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). *Analysis of recommendation algorithms for E-Commerce Proceedings of the ACM EC'00 conference, Minneapolis, MN* pp. 158–167.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). *Recommender systems in E-Commerce EC '99: Proceedings of the first ACM conference on electronic commerce, Denver, CO* pp. 158–166.

Schafer J. B., Konstan J. A., & Riedl J. (2001). *E-Commerce recommendation applications*. University of Minnesota.

Shardanand, U., & Maes, P. (1995). *Social information filtering: Algorithms for automating word of mouth Proceedings of the ACM CHI conference (CHI95)*.

Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: A system for sharing recommendations. *Communications of the ACM*, *40*(3), 59–62.

Wolf, J., Aggarwal, C., Wu, K., & Yu, P. (1999). *Horting hatches and egg: A new graph-theoretic approach to collaborative filtering Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining*.