9 July 2011

Collaborative Filtering With User Interest Evolution

Tsang-Hsiang Cheng
Southern Taiwan University, cts@mail.stut.edu.tw

Hung-Chen Chen
National Taiwan University, JesseHCChen@gmail.com

Wen-Ben Lin
National Tsing Hua University, d9534819@oz.nthu.edu.tw

Yen-Hsien Lee
National Chiayi University, yhlee@mail.ncyu.edu.tw

ISBN: [978-1-86435-644-1]; Full paper

COLLABORATIVE FILTERING WITH USER INTEREST EVOLUTION

Tsang-Hsiang Cheng, Department of Business Administration, Southern Taiwan University, Tainan, Taiwan, R.O.C., cts@mail.stut.edu.tw

Hung-Chen Chen, Department of Information Management, National Taiwan University, Taipei, Taiwan, R.O.C., JesseHCChen@gmail.com

Wen-Ben Lin, Institute of Engineering Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan, R.O.C., d9534819@oz.nthu.edu.tw

Yen-Hsien Lee, Department of Management Information Systems, National Chiayi University, Chiayi, Taiwan, R.O.C., yhlee@mail.ncyu.edu.tw

Abstract

Effective recommendation is indispensable to customized or personalized services. The ease of collecting, integrating and analyzing vast amounts of data about customers and their purchase intentions/behaviors concerning different products or services has greatly fostered the interest in automated recommendations appropriate for individual customers' needs, wants, or preferences. Collaborative filtering is a salient technique to support automated recommendations. However, the traditional collaborative filtering approach mainly relies on the assumption that all the given preferences are equally important, irrelevant of when a preference is collected. This assumption ignores the fact that a user's interests may be changed over time, and the prediction outcome of the traditional collaborative filtering approach may be misguiding if the preferences given at different time are not distinguished appropriately. Therefore, we propose a new collaborative filtering approach to take user interest evolution into account. Specifically, a clustering algorithm is first adopted to group the similar items. Subsequently, for a user, the preference of each cluster is calculated by the given preferences on each item in this cluster as well as the corresponding timestamps. A user's interest is then represented as a vector containing the preferences of all clusters. As a result, users with the most similar interest vectors to that of the active user will be chosen as his/her neighbors for collaborative recommendation. The experimental results demonstrate that our proposed approach improves the recommendation effectiveness in comparison with the traditional collaborative filtering approach.

Keywords: Recommendation, Collaborative Filtering, Time Weight, User Interest Evolution, Clustering.

1 INTRODUCTION

With the advances of the Internet, information overload has become a critical challenge facing individuals, giving rise to the need of providing recommendations for individuals. As the orientation of customization and personalization is highly praised, recommendation systems have emerged as a kind of e-service and will become a required service. Vendors also develop automated recommendation systems to better meet customers' needs or preference (Adomavicius and Tuzhilin, 2005). A highly celebrated success story is Amazon's personalized recommendation services, enabled by thorough analyses of voluminous browsing and purchase behaviors of customers.

Among various recommendation approaches, collaborative filtering is the most successful and widely adopted one (Sarwar et al., 2000) for supporting automated recommendations in various areas (e.g., filtering Usenet News, recommending TV shows and Web personalization (Ding et al., 2006)). Collaborative filtering, also known as social filtering or user-to-user correlation analysis, identifies users whose tastes are similar to those of an active user and recommends items they have liked (Herlocker et al., 1999; Herlocker et al., 2000; Konstan et al., 1997; Resnick et al., 1994; Shardanand and Maes, 1995). Specifically, by computing the similarity of users, a set of "nearest neighbor" users whose known preferences correlate significantly with those of an active user are first identified. The preference for unseen target item can then be predicted for the active user on the basis of the preferences of those nearest neighbors on the target item. Thus, in the collaborative filtering approach, users share their preferences regarding each item they purchased so that other users can better decide which items to consume (Ansari et al., 2000; Balabanovic and Shoham, 1997; Schafer et al., 2001).

However, the traditional collaborative filtering approach mainly relies on the assumption that all the given preferences are equally important, irrelevant of when a preference is collected. This assumption ignores the fact that a user's interests may be changed over time, and the prediction outcome of the traditional collaborative filtering approach may be misguiding if the preferences given at different time are not distinguished appropriately. For example, a man previously liked the suspense movies and provided lots preferences on these movies. Once he changes his interests as time goes by (e.g., start to see the comedy movies), the traditional collaborative filtering approach will suppose the man is still interested in the suspense movies until his preferences on the comedy movies are sufficient. In response, a preference given recently by a user should have greater impact on the recommendation predictions than another preference given a long time ago. Assigning different weights to the preferences given at different time can help to derive a user's current interests and the recommendation effectiveness can be further improved based on the evolutional interests.

Therefore, we propose a new collaborative filtering approach to take user interest evolution into account. Specifically, a clustering algorithm is first adopted to group the items with similar contents. Subsequently, for a user, the preference of each cluster is calculated by the given preferences on each item in this cluster as well as the corresponding timestamps. A user's interest is then represented as a vector containing the preferences of all clusters. As a result, users with the most similar interest vectors to that of the active user will be chosen as his/her neighbors for collaborative recommendation. We collect the movies from the MovieLens dataset and the corresponding contents from the International Movie Database (IMDB). Then, we conduct a series of experiments using the traditional collaborative filtering approach as the performance benchmark. The experimental results demonstrate that our proposed approach improves the recommendation effectiveness in comparison with the traditional collaborative filtering approach.

The remainder of this paper is organized as follows. In Section 2, we review the literature related to the collaborative recommendation approaches. In Section 3, the details of our proposed approach are presented. Section 4 reports the evaluation dataset, experimental design and the significant experimental results. We conclude in Section 5 with a summary and some future directions.

2 LITERATURE REVIEW

Among the existing recommendation systems, the collaborative filtering approach is the most successful and widely adopted one. The concept of the collaborative filtering approach is to utilize the opinions from other users who have the similar tastes to predict the preference on a target item for the active user (Balabanovic and Shoham, 1997; Herlocker et al., 1999). As shown in Figure 1, in a typical collaborative filtering recommendation scenario, there is a set of n users $U = \{u_1, u_2, \ldots, u_n\}$ and a set of m items $I = \{i_1, i_2, \ldots, i_m\}$. Each user u_i has a list of items I_{u_i} (where $I_{u_i} \subseteq I$ and I_{u_i} can be an empty set) on which the user has expressed his/her preferences. The preference of a user u_i on an item i_j (denoted as O_{ij}) can be a subjective rating explicitly stated by the user or an implicit measure inferred from purchase, browsing and navigation data in user activity. Regarding the recommendation decision itself, it can be made for a specific user u_a (where $u_a \in U$), called an active user, on those items that have not explicitly been rated or chosen by this user. Alternatively, it may suggest a new item i_{new} (where $i_{new} \notin I$) to those users who might be interested (Adomavicius and Tuzhilin, 2005; Sarwar et al., 2000).

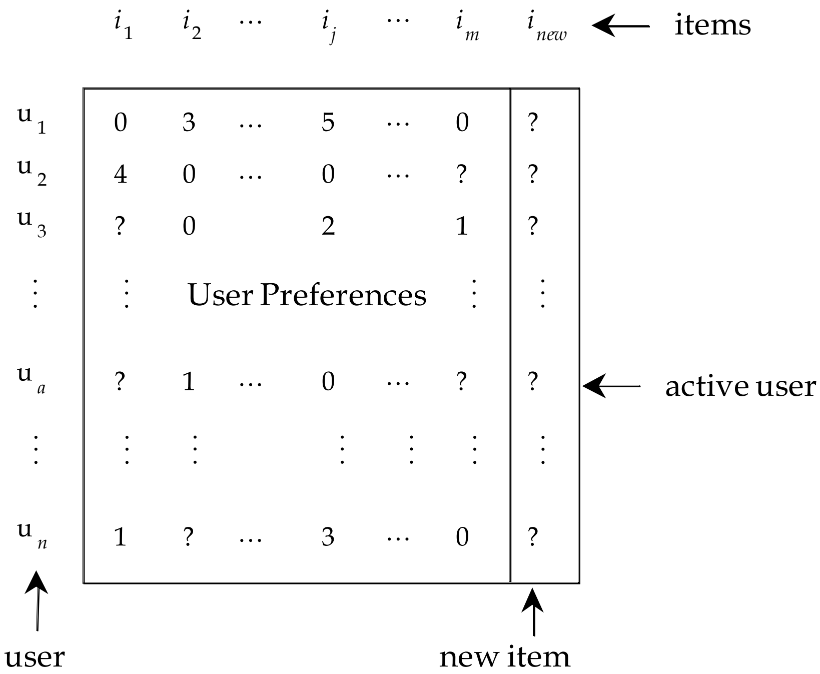

Figure 1. A typical collaborative filtering approach scenario.

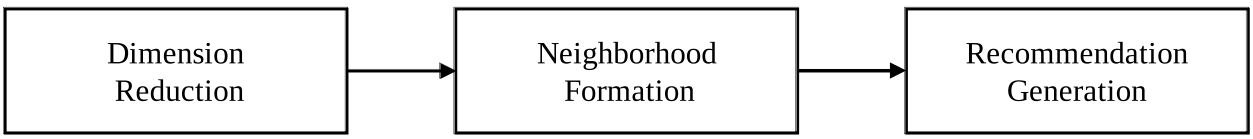

Figure 2. Process of collaborative filtering technique.

As shown in Figure 2, the process of a typical neighborhood-based (user-based) collaborative filtering approach can be divided into three phases (Sarwar et al., 2000; Wei et al., 20002):

1 Dimension Reduction: Transform the original user preference matrix into a lower dimensional space to address the sparsity and scalability problems.

2 Neighborhood Formation: For an active user, compute the user similarities between all other users and the active user and to form a proximity-based neighborhood with a number of like-minded

users for the active user. A number of similarity measures have been proposed to estimate the similarity between two users (Shardanand and Maes, 1995; Herlocker et al., 1999; Sarwar et al., 2000), including Pearson correlation coefficient, constrained Pearson correlation coefficient, Spearman rank correlation coefficient, cosine similarity, and mean-squared difference.

3 Recommendation Generation: After neighborhood formation, the collaborative filtering approach generates recommendations based on the preferences of the set of nearest neighbors of the active user. The deviation-from-mean method, adopted by GroupLens (Resnick et al., 1994; Konstan et al., 1997), is the most popular method for recommendation generation.

However, as we mentioned in Section 1, the traditional collaborative filtering approach mainly relies on the assumption that all the given preferences are equally important, irrelevant of when a preference is collected. This assumption may misguide the prediction outcome of the traditional collaborative filtering approach. There are only few researches have forced on the temporal features of the given preferences for making collaborative recommendations (Tang et al., 2003; Terveen et al., 2002, Zhao et al., 2005). However, the recency of preference has not been studied well (Ding and Li, 2005; Ding et al., 2006).

Ding and Li (2005) first presented an item-based collaborative filtering algorithm to take the changes in user purchases interests into account. Their main idea is to predict precisely user future purchases interests by deploying time weight. The item that was rated recently by a user is supposed to be much important than an item that was rated long time ago. Like the traditional item-based collaborative filtering algorithm, this algorithm first computes the similarity between two items by adopting the specific measure, such as cosine similarity, Pearson correlation coefficient, or conditional probability-based similarity. Subsequently, the prediction of the preference on a target item is computed by a modified weighted average method. Since Ding and Li (2005) assumed that the user purchase interest is sensitive to time, the modified weighted average method adopts a function *f(t)* to assign the weight for each involved preference, and the corresponding measure is defined as:

$$O_{ij} = \frac{\sum_{c=1}^{k} O_{ic} \times sim(i_j, i_c) \times f(t_{ic})}{\sum_{c=1}^{k} sim(i_j, i_c)},$$

where O_{ij} is the predicted preference of user u_i on item i_j, k is the number of nearest neighbors of item i_j, $sim(i_j,i_c)$ is the similarity between items i_j and i_c, and $f(t_{ic})$ is the time the preference O_{ic} was produced.

However, the algorithm (Ding and Li, 2005; Ding et al., 2006) does not consider the difference among users due to the essence of item-based collaborative filtering. Moreover, their algorithm does not consider the data sparsity problem (Burke, 2002) and item heterogeneities (Chen, 2010). In response, our proposed approach adopts the clustering algorithm to alleviate the data sparsity problem and tries to derive the user's current interests for identifying more appropriate neighbors. To avoid consulting the preferences on the irrelevant items, our proposed approach is also designed to only consider the preferences on items that similar to the target item for recommendation generation. Consequently, our proposed approach can achieve better prediction accuracy.

3 COLLABORATIVE FILTERING WITH USER INTEREST EVOLUTION (CFIE)

The traditional collaborative filtering approach mainly relies on the assumption that all the given preferences are equally important, irrelevant of when a preference is collected. However, as we mentioned in Section 1, it is not reasonable since a user's interests may be changed over time. Specifically, the preferences given at different time should have different weights. To catch a user's short-term interests without losing the long-term interests, we propose a novel collaborative filtering approach to consider user interest evolution. The proposed approach also concerns the data sparsity problem and item heterogeneities. As shown in Figure 3, there are four major phases in our proposed approach, i.e., item clustering, user interest derivation, neighborhood formation, and recommendation generation. The phase of item clustering is to group the items with similar contents by adopting a specific cluster algorithm. Subsequently, for each user, the phase of user interest derivation first calculates the preference of each item cluster based on the given preferences on each item in this cluster as well as the corresponding timestamps. Then, the interest vector of each user is constructed to represent the user's current interests. After user interest derivation, the phase of neighborhood formation is to identify the neighbors for the active user based on the similarity of the interest vectors. Finally, the recommendation generation phase estimate the preference for the active user on the target item based on the opinions of his/her neighbors. Note that the timestamp T_{ij} of a preference O_{ij} given by user u_i on item i_j is recorded in the involved databases for deriving user interests.

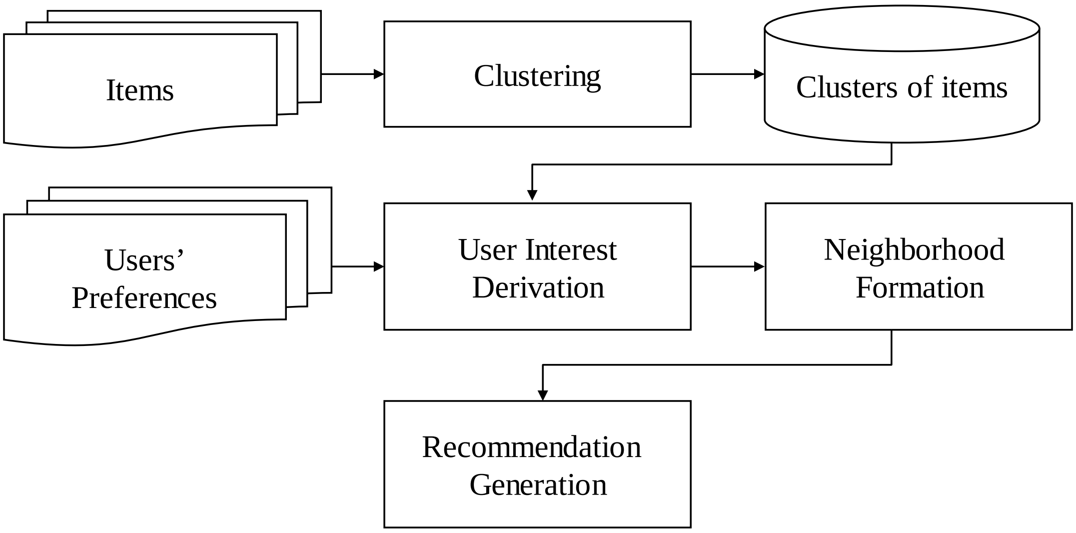

Figure 3. Overall process of our proposed approach.

3.1 Item Clustering

Because most of items to be recommended (e.g., movies, songs, or electronics) posses text descriptions, the text-processing tasks, i.e., keyword extraction, keyword selection, and item representation, can be performed in advance for estimating content similarities. For feature extraction, a set of nouns and noun phrases from the textual description of each item is first extracted. Subsequently, feature selection selects representative features from the feature set of all textual documents based on a chosen feature selection metric. Specifically, we employ TF×IDF as the feature selection method to select the top f keywords with the highest TF×IDF scores from the textual descriptions. Afterwards, each item is represented as a feature (i.e., keyword) vector composed by the f features with the corresponding TF×IDF scores. To assess the content similarity between two items, the cosine similarity measure is adopted. Then, we employ the famous k-means clustering algorithm to group the similar items into CN clusters. Note that CN is a pre-specified constant.

3.2 User Interest Derivation

In the phase of user interest derivation, we have to assess the preference of each item cluster for a user. The intuitive way is to summarize the preferences on the items in an item cluster as the corresponding cluster preference. However, this intuitive way does not take the interest evolution into account. Specifically, a preference given recently by a user should have greater impact on the recommendation predictions than another preference given a long time ago. To achieve the goal, we further consider two temporal factors, i.e., time weight and recency weight, to calculate the cluster preferences.

The time weight of a preference O_{ij} is defined as: $\frac{1}{\log(t_{ij})}$, where t_{ij} is the time difference between the current time T_{now} and the timestamp T_{ij} of O_{ij}. Accordingly, a preference given by a user a long time ago has a smaller time weight to reduce its impact on the cluster preference. Moreover, to catch the user's latest interest as soon as possible, we also introduce the recency weight w_r in our proposed approach. The recency weight is to promote the importance of the preferences given recently to deal with the situation that a user suddenly changes his/her interests. Given a pre-defined parameter t_c (i.e., time cut-point for recency weight), the recency weight of a preference O_{ij} is set to w_r (i.e., short-term preference weight) if $t_{ij} < t_c$ (t_{ij} is the time difference between the current time T_{now} and the timestamp T_{ij} of O_{ij}). Otherwise, the recency weight of a preference is set to 1- w_r (i.e., long-term preference weight).

Considering both of the time weight and the recency weight, the measure to calculate the preference of item class C_x for user u_i is defined as follows:

$$CP(u_i, C_x) = \sum_{i_j \in C_x} O_{ij} \times RW(i, j) \times \frac{1}{\log(t_{ij})},$$

Where $CP(u_i,C_x)$ is the preference of user u_i on item cluster C_x, and $RW(i,j)$ is the recency weight of the corresponding preference O_{ij}.

As a result, we can get the preferences of all item clusters for a user. The user interest of user U_i is then represented as a vector $\overrightarrow{IV_i}$ containing the preferences of all clusters. Specifically, the interest vector $\overrightarrow{IV_i}$ of user u_i is denoted as $<CP(u_i,C_1), CP(u_i,C_2), \ldots, CP(u_i,C_{CN})>$.

3.3 Neighborhood Formation

After user interest derivation, each user in the dataset has his/her personal interest vector. Note that a user's interest vector is dynamic and need to be updated once s/he provides a new preference. In our proposed approach, we adopt the popular cosine similarity measure to estimate the user similarity of two users u_a and u_b based on the corresponding interest vectors. The cosine similarity measure is defined as:

$$sim(u_a, u_b) = \frac{\overrightarrow{IV_a} \cdot \overrightarrow{IV_b}}{\left\|\overrightarrow{IV_a}\right\|_2 \times \left\|\overrightarrow{IV_b}\right\|_2} = \frac{\sum_{x=1}^{CN} CP(u_a, C_x) CP(u_b, C_x)}{\sqrt{\sum_{x=1}^{CN} CP(u_a, C_x)^2} \sqrt{\sum_{x=1}^{CN} CP(u_b, C_x)^2}},$$

Where $\overrightarrow{IV_a}$ (or $\overrightarrow{IV_b}$) is the interest vector of u_a (or u_b), $CP(u_a, C_x)$ (or $CP(u_b, C_x)$) is the preference of item class C_x for user u_a (or u_b), and CN is the number of item clusters.

Afterwards, the top N users with the highest user similarities estimated by this cosine similarity measure based on interest vectors are selected as the neighbors for the active user.

3.4 Recommendation Generation

After identifying the top N nearest neighbors for the active user u_a, the known preferences of the neighbors on the target item i_t are aggregated to arrive at a preference the prediction for u_a on i_t. We adopt the prevalent deviation-from-mean method (Konstan, et al., 1997; Resnick et al., 1994) by replacing the average preference of u_i with the average preference in item cluster C_x of u_i, and the modified measure is defined as follows:

$$O_{at} = \overline{O}(u_a, C_x) + \frac{\sum_{b=1}^{N} \left(O_{bt} - \overline{O}(u_b, C_x)\right) \times sim(u_a, u_b)}{\sum_{b=1}^{N} sim(u_a, u_b)},$$

Where O_{at} is the predicted preference of active user u_a on target item i_t, C_x is the item cluster that item i_t belongs to, and $\overline{O}(u_a, C_x)$ (or $\overline{O}(u_b, C_x)$) is the average preference in item cluster C_x of u_i (or u_b).

4 EMPIRICAL EVALUATION

We conduct the empirical evaluation of the proposed collaborative approach that considers user interest evolution and implement the traditional collaborative filtering approach as the performance benchmark. The evaluation dataset are first depicted in Section 4.1. The evaluation procedure and performance criteria are then presented in Section 4.2. Subsequently, the tuning experiments on the effects of related parameters for the two approaches are provided in Section 4.3. Finally, the comparative performance of the two approaches is presented in Section 4.4.

4.1 Data Collection

We use the MovieLens dataset collected by the GroupLens Research Project at the University of Minnesota to conduct a series of experiments. There are 100,000 ratings (with a scale from 1 to 5) from 943 users on 1,682 movies (from 20 Sep., 1997 to 22 Apr., 1998). All of the users in the original dataset have rated at least 20 movies. Furthermore, because the MovieLens dataset does not contain the description of each movie, we obtain these descriptions from the Internet Movie Database (IMDB) and represent each movie as a feature (keyword) vector accordingly (as introduced in Section 3.1).

4.2 Evaluation Procedure and Criteria

In the collected dataset, we use the preferences given before 11 Jan., 1998 as the training dataset (namely $D_{training}$) and the remaining preferences as the testing dataset (namely $D_{testing}$). There are about 80,000 preferences in $D_{training}$ and 20,000 preferences in $D_{testing}$. The preferences in $D_{training}$ are regarded as the given preferences and the preferences in $D_{testing}$ are used for the preference prediction tasks. Moreover, to avoid the possible bias, the following experiments are performed 30 times by randomly selecting 80% preferences from $D_{training}$ as the given preferences and 80% preferences from $D_{testing}$ for predictions to get the average performance.

Furthermore, we adopt the evaluation criteria, i.e., MAE, to evaluate the prediction accuracy of our proposed approach (namely CFIE) and the traditional collaborative filtering approach (namely CF). MAE is widely adopted measure to evaluate the prediction accuracy and is defined as the average absolute difference between the predicted ratings and the actual ratings as: $\text{MAE} = \frac{\sum_{j=1}^{T} |p_i - q_i|}{T}$, where p_i is a predicted preference score, q_i is its actual score for the same preference prediction task, and T is the number of preference prediction tasks.

4.3 Parameters Tuning Results

Our proposed approach, i.e., CFIE, involves several parameters, including the number of item cluster (CN), the weight for short-term preference (w_r), the cut-point (t_c) for defining the short-term preference, and the number of neighbors (N) in the neighborhood for prediction. Therefore, we conduct a series of tuning experiments to determine the appropriate values for the parameters involved in our proposed approach.

We first examined the effects of the number of item clusters *CN* for CFIE on MAE, ranging from 3 to 10 in increment of 1, on the prediction accuracy, by given the weight for short-term preference as 0.9 (i.e., w_r = 0.9), the cut-point for defining the short-term preference as 4 weeks (i.e., t_c = 4 weeks), and the number of neighbors as 10 (i.e., *N* = 10). As shown in Figure 4, the larger *CN* is, the worse performance CFIE achieves (when *CN* is greater than 4). Specifically, CFIE achieve the best prediction accuracy (i.e., MAE = 9.764) when *CN* is set to 4. Accordingly, we set *CN* to 4 for CFIE in the following experiments.

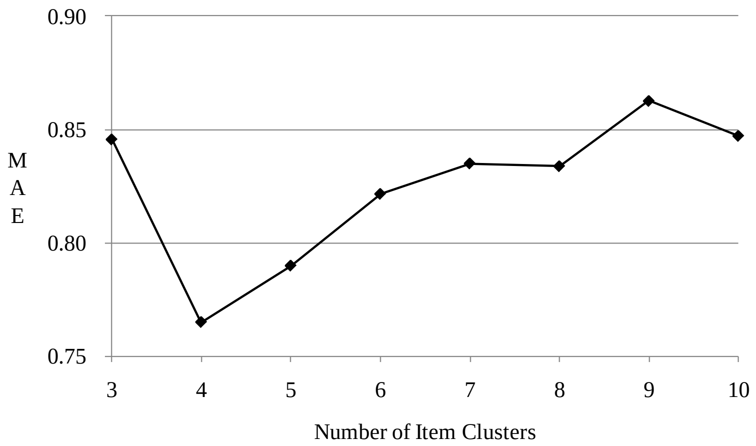

Figure 4. Effects of the number of item clusters for CFIE on MAE.

Subsequently, we examined the effects of the weight for short-term preference w_r for CFIE on MAE, ranging from 0.6 to 0.9 in increment of 0.1, on the prediction accuracy, by given the number of item clusters as 4 (i.e., *CN* = 4), the cut-point for defining the short-term preference as 4 weeks (i.e., t_c = 4 weeks), and the number of neighbors as 10 (i.e., *N* = 10). As shown in Figure 5, the larger w_r is, the better prediction accuracy CFIE achieves. Accordingly, we set w_r to 0.9 for CFIE in the following experiments.

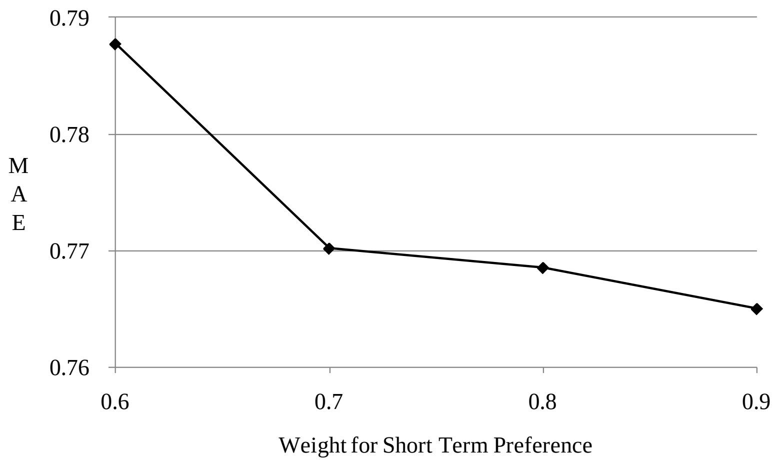

Figure 5. Effects of the weight for short-term preference for CFIE on MAE.

After determining the appropriate values for the parameters *CN and* w_r (i.e., *CN* = 4 and w_r = 0.9), we then examined the effects of the number of neighbors *N* for CFIE on MAE, ranging from 4 to 10 in increment of 2. Note that the cut-point for defining the short-term preference is still set to 4 weeks. As shown in Figure 6, the larger *N* is, the worse prediction accuracy CFIE achieves. Accordingly, we set *N* to 4 for CFIE in the following experiments.

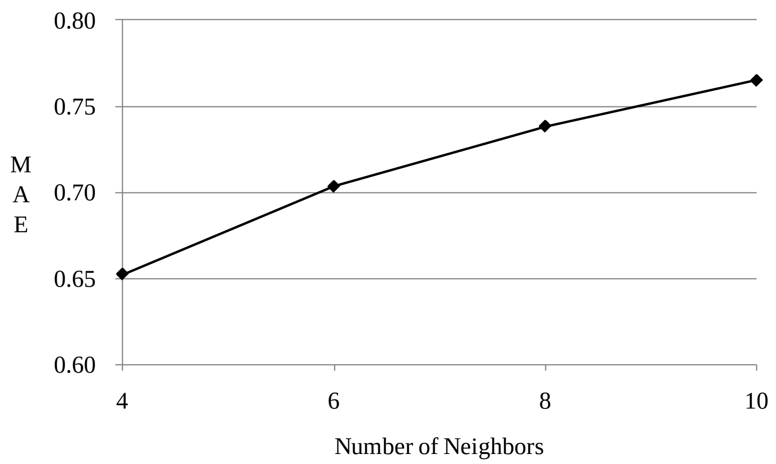

Figure 6. Effects of the number of neighbors for CFIE on MAE.

Finally, we examined the effects of the cut-point for defining the short-term preference t_c for CFIE on MAE, ranging from 1 weeks to 8 weeks in increment of 1 week, by given the number of item clusters as 4 (i.e., *CN* = 4), the weight for short-term preference as 0.9 (i.e., w_r = 0.9), and the number of neighbors as 10 (i.e., *N* = 4). As shown in Figure 7, the prediction accuracy of CFIE increases when we increase t_c from 1 week to 3 weeks. However, the prediction accuracy of CFIE decreases when t_c is more than 3 weeks. Therefore, we set t_c to 3 weeks (MAE = 0.649) in the following experiments.

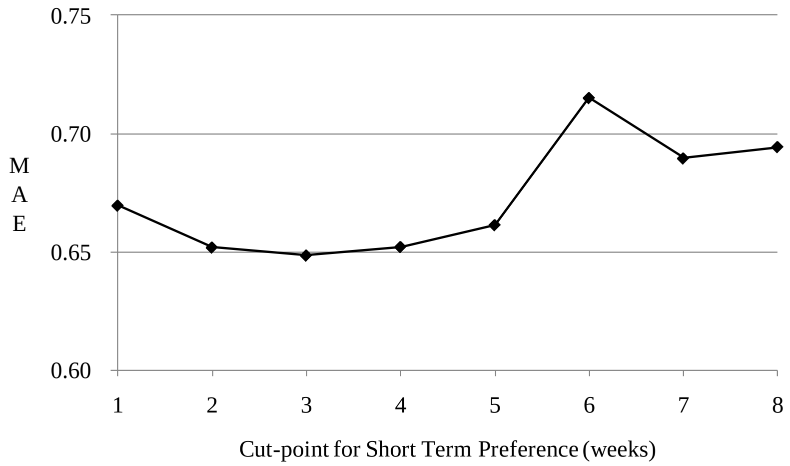

Figure 7. Effects of cut-point of short-term preference for CFIE on MAE.

Since the number of neighbors is also a parameter of the performance benchmark, i.e., CF, we also examined the effects of the number of neighbors *N* for CF on MAE, ranging from 4 to 10 in increment of 2. As shown in Figure 8, the larger *N* is, the worse prediction accuracy CF achieves. Accordingly, we also set *N* to 4 for CF in the following experiments.

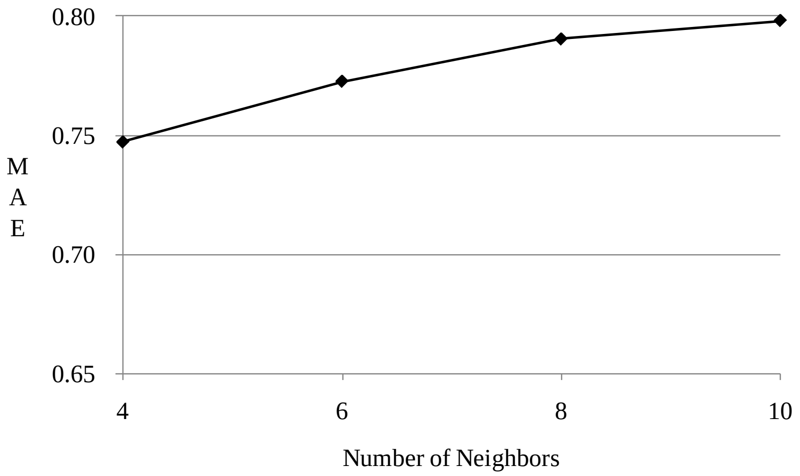

Figure 8. Effects of the number of neighbor for CF on MAE.

4.4 Comparative Evaluation Results

According to the parameter tuning experimental results displayed in Section 4.3, we summarize the final settings for the parameters of CFIE and CF in Table 1. Since both of the numbers of neighbors are set to 4 for CFIE and CF, we can fairly compare the prediction accuracy of the two approaches under the same condition. The comparative evaluation results are shown in Table 2. The results show that CFIE significantly outperforms CF on MAE. In summary, the results suggest that our proposed approach can catch the updated user interests by considering both of the temporal factors, i.e., the time weight and the weight for short-term preference. Unlike the traditional collaborative filtering approach, our proposed approach tries to distinguish the importance of the preferences given at different time such that the recommendation prediction would not be seriously misguiding based on the preferences that may be out of date. Specifically, our proposed approach can keep the short-term and the long-term interests simultaneously for the collaborative recommendations, and therefore has better prediction accuracy. Moreover, the proposed approach also considers the item heterogeneities such that the preferences on the irrelevant items (i.e., the item with dissimilar content) will not be utilized for preference prediction. As a result, our proposed approach avoids utilizing the unreliable preferences that may decrease the prediction accuracy.

	Number of Item Clusters (*CN*)	Weight for Short-Term Preference (w_r)	Number of Neighbors (*N*)	Cut-Point for Short-Term Preference (t_c)
CFIE	4	0.9	4	3 (weeks)
CF	—	—	4	—

Table 1. Summary of the parameter settings for CFIE and CF.

Recommendation Approach	Prediction Accuracy (MAE)
CFIE	0.649***
CF	0.747

***:p<0.01

Table 2. Comparative evaluation results of CFIE and CF on MAE.

5 CONCLUSION

Among various kinds of recommendation approaches, the collaborative filtering is the most successful and widely adopted one. However, the traditional collaborative filtering approach mainly relies on the assumption that all the given preferences are equally important, irrelevant of when a preference is collected. This assumption ignores the fact that a user's interests may be changed over time, and the prediction outcome of the traditional collaborative filtering approach may be misguiding if the preferences given at different time are not distinguished appropriately. In response, we proposed a novel cluster-based collaborative approach that takes user interest evolution into account. Specifically, our proposed approach tracks the short-term and the long-term interests simultaneously for identifying reliable neighbors. Moreover, our proposed approach is also designed to alleviate the sparsity problem and consider item heterogeneities for better performance. The experimental results demonstrate that our proposed approach outperforms the traditional collaborative filtering approach under the same condition.

Other ongoing and future research directions are briefly discussed as follows. First, to improve the generalizability of the evaluation results in this report, we should conduct additional evaluations that involve different contexts (i.e., book, music recommendations) in the future. Second, our proposed approach adopts the function $\frac{1}{\log(t_{ij})}$ to define the time weight. Other possible functions could be investigated for better performance. Finally, different user may prefer different parameter settings. For example, some users may never change their interests, but some users may be fickle all the time. To automatically choose the appropriate parameter settings for each user will be an essential direction for future research.

ACKNOWLEDGEMENT

This work was supported by the National Science Council of the Republic of China under the grant NSC 97-2410-H-218-017-MY2, NSC 98-2410-H-002-232-MY2, and NSC 99-2410-H-218-022 -MY2.

References

Adomavicius, G., and A. Tuzhilin (2005). Toward 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.

Alspector, J., A. Kolcz, and N. Karunanithi (1998). Comparing Feature-based and Clique-based User Models for Movie Selection. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, 11-18.

Ansari, A., S. Essegaier, and R. Kohli (2000). Internet Recommendation Systems. *Journal of Marketing Research*, (37), 363-375.

Balabanovic, M. and Y. Shoham (1997). Fab: Content-based, Collaborative Recommendation. *Communications of the ACM*, 40(3), 66-72.

Burke, R., (2002). Hybrid Recommendation Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331-370.

Chen, H. C., C. P. Wei, Y. S. Dai, and Y. K. Lin (2010). Exploiting Item Heterogeneity for Collaborative Filtering Recommendation. In *Proceedings of the 4th China Summer Workshop on Information Management.*

Ding, Y., and X. Li (2005). Time Weight Collaborative Filtering. In *Proceeding of Information and Knowledge Management*, 485-492.

Ding, Y., X. Li, and M. E. Orlowska (2006). Recency-Based Collaborative Filtering. In *Proceedings of the 17th Australasian Database Conference*, 99-107.

Herlocker, J., J. Konstan, and J. Riedl (2000). Explaining Collaborative Filtering Recommendations. In *Proceeding of ACM 2000 Conference on Computer Supported Cooperative Work*, 241-250.

Herlocker, J., J. Konstan, A. Borchers, and J. Riedl (1999). An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 230-237.

Konstan, J., B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3), 77-87.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews, In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)*, 175-186.

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl (2000). Analysis of Recommendation Algorithms for E-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 158-167.

Schafer, J., J. Konstan, and J. Riedl (2001). E-commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5, 115-153.

Shardanand, U., and P. Maes (1995) Social Information Filtering: Algorithms for Automating "Word of Mouth", In *Proceedings of ACM Conference on Human Factors and computing Systems*, 210-217.

Tang, T. Y., P. Winoto, and K. C. C. Chan (2003). On the Temporal Analysis for Improved Hybrid Recommendations. In *Proceedings of International IEEE/WIC Conference on Web Intelligence*, 214-220.

Terveen, L., J. McMackin, B. Amento, and W. Hill (2002). Specifying Preferences Based on User History. In *Proceedings of Conference on Human Factors in Computing Systems*, 315-322.

Wei, C. P., M. Shaw, and R. Easley (2002). A Survey of Recommendation Systems in Electronic Commerce. Chapter 8 in *E-Service: New Directions in Theory and Practice*, R. T. Rust and P. K. Kannan (Eds), M. E. Sharpe Publisher.

Zhao, Y., C. Zhang, and S. Zhang (2005). A Recent-Biased Dimension Reduction Technique for Time Series Data. In *Advances in Knowledge Discovery and Data Mining: Lecture Notes in Computer Science*, 107-176.