

Association for Information Systems AIS Electronic Library (AISeL)

PACIS 2014 Proceedings

Pacific Asia Conference on Information Systems
(PACIS)

2014

IMPROVING RECOMMENDATION PERFORMANCE WITH USER INTEREST EVOLUTION PATTERNS

Tsang-Hsiang Cheng

Southern Taiwan University of Science and Technology, cts@stust.edu.tw

Yen-Hsien Lee

National Chiayi University, yhlee@mail.ncyu.edu.tw

Follow this and additional works at: <http://aisel.aisnet.org/pacis2014>

Recommended Citation

Cheng, Tsang-Hsiang and Lee, Yen-Hsien, "IMPROVING RECOMMENDATION PERFORMANCE WITH USER INTEREST EVOLUTION PATTERNS" (2014). *PACIS 2014 Proceedings*. 298.

<http://aisel.aisnet.org/pacis2014/298>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2014 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

IMPROVING RECOMMENDATION PERFORMANCE WITH USER INTEREST EVOLUTION PATTERNS

Tsang-Hsiang Cheng, Department of Business Administration, Southern Taiwan University of Science and Technology, Tainan, Taiwan, R.O.C., cts@stust.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. Collaborative filtering approach is a salient technique to support automated recommendations, which relies on the profiles of customers to make recommendations to a target customer based on the neighbors with similar preferences. However, traditional collaborative recommendation techniques only use static information of customers' preferences and ignore the evolution of their purchasing behaviours which contain valuable information for making recommendations. Thus, this study proposes an approach to increase the effectiveness of personalized recommendations by mining the sequence patterns from the evolving preferences of a target customer over time. The experimental results have shown that the proposed technique has improved the recommendation precision in comparison with collaborative filtering method based on Top k recommendation.

Keywords: Sequential Pattern, Collaborative Filtering, Customer Preference Profile

1 INTRODUCTION

Internet has become an important way to acquire information nowadays. However, the selectable information exceeds the cognitive load capacity of web users, that is to say, the rapid growth of web has also brought the problem of information overload (Häubl and Trifts, 2000). In order to reduce the search cost of information overload, a variety of collaborative filtering algorithms have been proposed to facilitate the development of automatic recommendation systems (Wan et al., 2003).

Since e-commerce market has been fast-developing, personalized recommendation service is absolutely essential for the customer-focused and personalized websites (Adomavicius and Tuzhilin, 2005; Schafer et al., 2001). Recommendation systems would identify customers' potential needs from their search behaviors and historical preference records, thereby offering customers appropriate personalized recommendations. The valuable recommendations not only effectively help users solve the information overload problem but also help websites provide personalized service accordingly. As a result, the high quality recommendation services have become a powerful tool and a critical strategy on the relationship marketing for e-commerce (Strasser and Zugenmaier, 2003; Wei et al., 2005).

Content-based recommendation and Collaborative filtering recommendation are the two most popular techniques. Content-based approach utilizes a series of features related to product descriptions. And collaborative filtering makes use of product preferences and similar user evaluations (i.e., neighbor users) (Beyah et al., 2003; Balabanovic and Shoham, 1997; Konstan et al., 1997; Mooney and Roy, 1999). Content-based recommendation analyzes the content descriptions of a customer's favored products, and then creates a product preference profile to provide appropriate recommendations in the future. On the other hand, collaborative filtering recommendation firstly searches for a group of users with similar product preferences for a target customer, and then integrates product evaluation of the group to provide prediction of product preference for the target customer (Adomavicius and Tuzhilin, 2005; Lang, K. 1995; Sarwar et al., 2000; Wei et al., 2002). Collaborative filtering technique ignores the dimension of time and assumes that the information of customer preference is static (Khatri, 2012). However, a customer's interest may change over time which implies a type of dynamic information (Cho et al., 2005; Min and Han, 2005; Ding et al., 2006). Therefore, Cho et al. organized a customer's product items in a sequential order to present how one's buying behavior changes, and accordingly make recommendation with sequential rule (Cho et al., 2005). Besides, Min and Han proposed a method to detect the changes in customers' preferences based on clustering technique. They induced the preference changes from a group of users with similar interests to increase the accuracy of recommendation (Min and Han, 2005). In addition, some scholars suggested that the evolution pattern of customers' interests should be added into current recommendation techniques to improve the recommendation performance (Hijikata et al., 2009; Wei et al., 2009).

Therefore, we proposed a sequential pattern mining-based technique, which takes evolution pattern of user's interest into account. A clustering algorithm is specifically adopted to group the items with similar contents. Subsequently, for a target user, the preference rating of each item cluster is the average rating in the cluster at the corresponding timestamps. A user's interest is then represented as a vector using the preferences of all clusters. As a result, the generalized Sequential Patterns Mining (GSP) method is used to generalize customers' interest evolution patterns for making recommendations. We collect movies from the MovieLens dataset and 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 could considerably improve 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 literatures related to recommendation techniques and sequential pattern mining. In Section 3, the details of our proposed approaches are presented. Section 4 reports the evaluation dataset, experimental design and the significant experimental results. We make a summary and some future directions in Section 5.

2. LITERATURE REVIEW

2.1 Recommendation Technique

Collaborative filtering is one of the most successful techniques for recommender systems. The major concept of collaborative filtering is to utilize the opinions from other users who have the similar tastes to predict the preference on a target item for an active user (Balabanovic and Shoham, 1997; Herlocker et al, 2004). As shown in Figure 1, in a typical collaborative filtering recommendation scenario, there is a set of n users $U = \{u_1, u_2, \dots, u_n\}$ and a set of m items $I = \{i_1, i_2, \dots, i_m\}$. 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 or her preferences. The preference of a user u_i for an item i_j (denoted as O_{ij}) can be a subjective rating explicitly stated by the user or an implicit measure inferred from the user's purchased items, browsing and navigation behaviors. The recommendation can be made for a specific user u_a (where $u_a \in U$) on those items that have not explicitly been rated or chosen by the user. Alternatively, the recommender system 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).

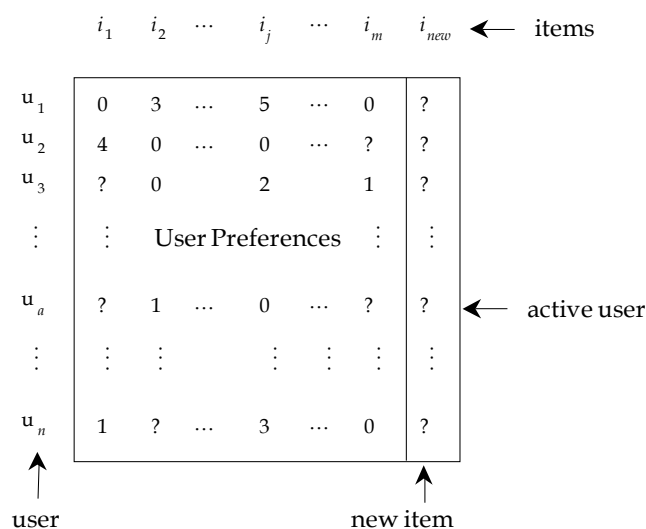


Figure 1. User-item rating matrix used for collaborative filtering approach

As shown in Figure 2, the process of a typical neighborhood-based (user-based) collaborative filtering algorithm can be divided into three phases (Sarwar et al 2000; Wei et al., 2002):

- 1 Dimensional Reduction: The process of dimension reduction transforms the original user-item rating matrix into a lower dimensional space to address the sparsity and scalability problems.
- 2 Neighborhood Formation: This step computes the similarities between an active user and all other users to form a proximity-based neighborhood with a number of like-minded users for the active user. There is a few number of similarity measure techniques (Herlocker et al, 1999; Sarwar et al., 2000), such as Pearson correlation coefficient, constrained Pearson correlation coefficient, Spearman rank correlation coefficient, cosine similarity, and mean-squared difference.
- 3 Recommendation Generation: It generates recommendations based on the preferences of the set from nearest neighbors of an active user. The deviation-from-mean method, adopted by GroupLens (Konstan et al, 1997), is the most popular method for making recommendation.

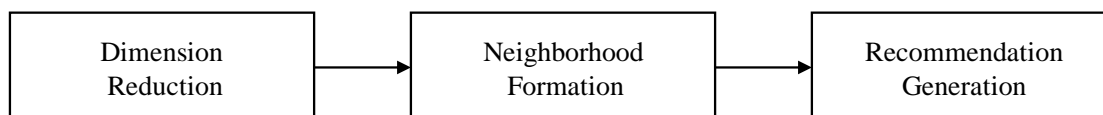


Figure 2. Overall process of collaborative filtering technique

As we mentioned in Section 1, the traditional collaborative filtering approach mainly relies on the assumption that all given preferences are equally important. However, it ignores the time dimension once a preference is collected. This assumption may misguide the prediction outcome of the traditional collaborative filtering approach. Not many researches have focused on the temporal features of the given preferences for making collaborative recommendations (Tang et al., 2003; Terveen et al., 2002; Zhao et al., 2005).

Ding and Li (2005) first presented an item-based collaborative filtering algorithm that takes the changes of user's purchase interests into account. Their main idea is to predict user future purchase interests precisely by deploying time weight. The item which has been rated recently by a user is supposed to be much more important than an item has been rated long time ago. Like the traditional item-based collaborative filtering algorithm, this algorithm first computes the similarity between two items by a 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 time sensitive, the modified weighted average method adopts a function $f(t) = e^{-\lambda t}$ to assign the weight for each involved preference, and the corresponding measure is defined as:

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

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 , $\text{sim}(i_j, i_c)$ is the similarity between items i_j and i_c , and $f(t_{ic})$ is the time the preference O_{ic} was produced.

Ding et al. (2006) redesigned an exponential weight W_c to replace the original time weight function $f(t)$ because they hope to simplify the parameter λ estimates of function $f(t)$. The exponential weight W_c is given as: $(1 - \frac{|O_{ic} - O_{ir}|}{|R|})^\alpha$, where O_{ic} is the preference of user u_i on item i_j , O_{ir} is the customer u_i 's latest product rating record for product i_c in the cluster, $|R|$ is the range of product rating (e.g., it's a Likert 5-level scale, then $|R|$ is 5), and α is a parameter of time weight. After induction of W_c , the weighted average method which is in use of measuring a target customer's preference would be modified as:

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

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 , $\text{sim}(i_j, i_c)$ is the similarity between items i_j and i_c , and $f(t_{ic})$ is the time the preference O_{ic} was produced.

Cho et al. (2005) argued that existing collaborative filtering systems are static, and the information of the purchase sequences of customers has been thus ignored. Therefore, they proposed a new method to enhance the performance of CF recommendation through uses of customer purchase sequences (Cho et al., 2005). As for operation procedure, they rearranged a customer profile considering the transaction time and a SOM (Self Organizing Map) clustering technique is employed to generate the "customer transaction cluster" $C = \{C_1, C_2, \dots, C_q\}$. The change in the transaction, clusters represents a customer purchase sequence. Therefore, observing the changes in the transaction clusters of each

customer over time, a purchase sequence of which can be built for each customer, and the purchase sequences are potentially capable of predicting the future purchase interest from a target customer. Cho et al. (2005) adopted association rules to extract the most frequent from customers' purchase sequence with the form of $R_j: r_{j,T-l+1}, \dots, r_{j,T-l} \Rightarrow r_{j,T}$. When making recommendations for the target customer u_i at the timestamp T , we can find the best matched dynamic behavior $R_j^c = \langle r_{T-l+1}, \dots, r_{j,T-l} \rangle$ in the purchase sequence database with the customer u_i 's buying behavior profile at the time $T-l$, $L_i^c = \langle C_{i,T-l+1}, \dots, C_{i,T-l} \rangle$. The found best-fit behavior rule $r_{j,T-l+1}, \dots, r_{j,T-l} \Rightarrow r_{j,T}$ can help determine the products the target customer u_i would most likely to buy.

2.2 Sequential pattern mining

Sequential pattern is a sequence of itemsets that frequently occurred in a specific order, all items in the same itemsets are supposed to have the same transaction-time value or to be within a same time gap. Usually a serial transactions of a customer are viewed as a sequence, usually called customer-sequence. In a customer-sequence S , each transaction is represented as an itemset in the sequence and all the transactions are listed in a certain order regarding to the transaction-time. For example, if the given transaction-itemsets are (a_1, a_2) , (b) and (c) , the customer-sequence S will be represented as: $\langle (a_1, a_2), (b), (c) \rangle$. Using customer-sequence to describe customer behavior may help e-shops apply recommendation mechanism to increase customers' service satisfaction or the overall sales performance.

Sequential pattern mining is a process of extracting certain sequential patterns whose support exceeds a predefined minimal support threshold (Bettin et al., 1998; Zhao et al., 2003). Since the number of sequences can be very large, and users may have different interests and requirements, to get the most interesting sequential patterns, usually a minimum support is pre-defined by users. By using the minimum support, we can prune those uninteresting sequential patterns, and thereby make the mining process more efficient. Obviously a higher support of sequential pattern is desired for more useful and interesting sequential patterns. Most sequential pattern mining algorithms are based on the Apriori property proposed in association mining; the property states that any sub-pattern of a frequent pattern must be frequent. AprioriAll, AprioriSome, DynamicSome (Agrawal and Srikant, 1995), GSP (Srikant and Agrawal 1996) and SPADE (Zaki, 2001) are the series of Apriori-like algorithms. The representative Apriori-like algorithm was GSP (Generalized Sequential Patterns) proposed by Srikant and Agrawal (Srikant and Agrawal, 1996). GSP integrates with time constraints and relaxes the definition of transaction. For time constraints, maximum gap and minimal gap are defined to specify the gap between any two adjacent transactions in the sequence. If the distance is not in the range between maximum and minimal gap, the two transactions cannot be taken as two consecutive ones in a sequence. This algorithm relaxes the definition of transaction by using a sliding window, which means if the distance between the maximal transaction time and the minimal transaction time of those items is no bigger than the sliding window then those items can be taken as in the same transaction.

GSP contains two main sub-processes: candidate pattern generation and frequent pattern generation. In the candidate generation process, candidate k -sequences are generated based on the large $(k-1)$ -sequences (Bettini et al., 1998; Mannila et al., 1995; Srikant and Agrawal, 1996; Zhao and Bhowmick, 2003). GSP performs relatively better than AprioriAll, in GSP the number of candidate sequences is much smaller. In addition, time constraints, taxonomies are integrated during the sequential patterns mining process to produce more knowledge (Pei et al, 2001; Rolland, 2001; Zaki, 2001; Zhao and Bhowmick, 2003). Therefore, to implement the sequential pattern mining task for extracting the dynamic profile on customer preferences, this study adopted the sequential pattern mining tool based on GSP algorithm provided by Weka platform (Hall et al, 2009).

3 RECOMMENDATION TECHNIQUE WITH USER INTEREST EVOLUTION (SPIE)

The traditional collaborative filtering mainly relies on the assumption that all the given preferences are equally important. However, the time dimension is ignored once a preference is collected. The traditional approach only considers users' newest preferences and overlooks the other occurred preferences. As we mentioned in Section 1, user's interests may be changed over time and the preferences given at different time could provide the information of users' preference evolution. To catch the users' interest patterns, we propose a novel recommendation approach (SPIE) with the sequential pattern mining technique which takes user interest evolution into account. The proposed approach also concerns the data sparsity problem and item heterogeneities. As shown in Figure 3, there are three major phases in the proposed approach, i.e., item clustering, preference templates construction, and template selection & recommendation generation.

The first phase of item clustering is to group the items with similar contents by adopting a specific cluster algorithm. Next, the phase of preference template construction first estimates the favorite status of each item-cluster for each user according to the given item preferences, and then the sequential pattern mining technique, GSP, is employed to construct users' dynamic preference templates for recommendation. Finally, the third phase finds the best-fit preference profiles for the active user according to his/her interest change and generates the recommendation contents.

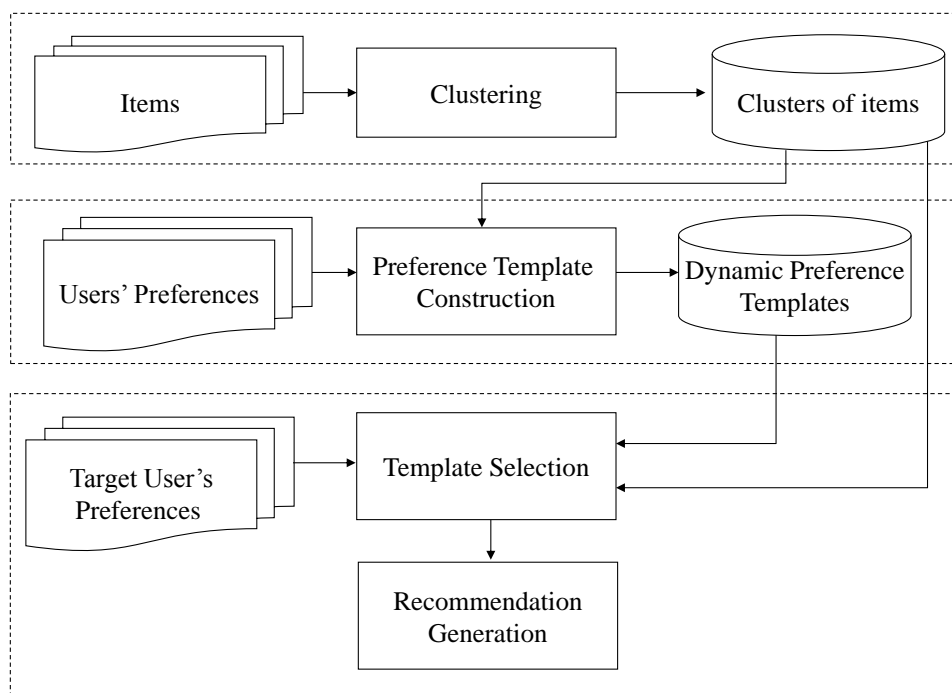


Figure 3. Overall structure of our proposed technique

3.1 Item Clustering

We assumed that customers have same preferences for similar products, and item clusters were constructed to simplify complexity of sequential pattern mining on recommendations. Due to the fact that all of items to be recommended can be described with text content (plot keywords from The Internet Movie DataBase (IMDB)), the text-processing tasks (i.e., keyword extraction, keyword selection, and item representation) can be performed in advance to estimate content similarities (Larsen and Aone, 1999; Roussinov and Chen, 1999). For feature extraction process, 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 a 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 top- f features with the corresponding TF×IDF scores (Pazzani and Billsus, 1997). 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 n clusters, and n is a pre-specified constant.

3.2 Preference Template Construction

Since the changes of customers' interest were retained in preferences at different timestamps, we employed sequential pattern mining method to analyze the preference records to discover the change patterns of customers' preference to construct "the dynamic preference templates". With the constructed templates, we can understand the changes in customers' interests and predict their purchase tendency in the future in order to help make recommendations. This phase contains three tasks: (1) Generation of summarized preference matrix, based on item clusters at different timestamps, (2) Transformation of the summarized preferences for mining, (3) Generation of the dynamic preference templates. The details of the three tasks are described as follows:

- (1) Generation of summarized preference matrix: with clustering technique, all items are clustered into r clusters C_1, C_2, \dots, C_r . Then, the original customer preference matrixes, as shown in figure 4, should be transformed into summarized preference matrix on item clusters as shown in figure 5. In the figure 5, the summarized preference of customer u_a 's on cluster C_k at timestamp T_i is described as $r_{a,t,k}$ and the value of $r_{a,t,k}$ is estimated as the average value of u_a 's preference for all the items in the cluster C_k . The calculation of preference $r_{a,t,k}$ is defined as follows:

$$r_{a,t,k} = \sum_{i \in C_k} \frac{1}{RN_k} (r_{a,i}),$$

where RN_k is the number of the items in the cluster C_k , $r_{a,i}$ is u_a 's preference value for the item i at the timestamp t .

	C_1						C_k		C_r		
	i_1	i_2	i_3	...	i_7	i_8	i_j	i_m
T_1	2	5	...			5			4
T_2		3		...		4				2
T_3	2		1	...		1			3	4
.		
.
.		

Figure 4. Original preference matrix for customer u_a

	C_1	C_2	C_3	...	C_{r-1}	C_r
T_1	$r_{a,1,1}$	$r_{a,1,2}$	$r_{a,1,3}$	$r_{a,1,r}$
T_2	$r_{a,2,1}$	$r_{a,2,2}$	$r_{a,2,3}$	$r_{a,2,r}$
T_3	$r_{a,3,1}$	$r_{a,3,2}$	$r_{a,3,3}$	$r_{a,3,r}$
.	.	.				
.
.	.	.				

Figure 5 Summarized preference matrix for customer u_a

- (2) Transformation of the summarized preferences for mining: traditional collaborative recommendation method considers every rating from customers is the same weight regardless of values of the ratings. However, Mobasher et al. (2001) proposed that recommendation systems should pay more attention to items with higher ratings when making recommendations (Mobasher et al., 2001). Symeonidis et al. (2008) also suggests that simply by using items with high ratings would increase recommendation performance. Thus, we transformed the rating $r_{a,t,k}$ in the summarized preference matrix as shown in figure 5 into a binary value scale (1 for like; 0 for others) according a threshold 2.5 (for the Likert 5-level scale). Then, all the elements of summarized preference matrix of figure 5 are transformed as binary value for sequential pattern mining.
- (3) Generation of the dynamic preference templates: After the mentioned matrix transformations, we adopted a sequential pattern mining module on Weka platform, according to GSP sequential pattern mining algorithm proposed by Srikant and Agrawal (Srikant and Agrawal, 1996), to mine the dynamic preference templates for recommendation.

3.3 Template Selection and Recommendation Generation

This phase also includes three tasks: (1) Creation of target user's interest evolution from his/her existing preferences, (2) Search for the best-fit dynamic preference profiles, (3) Generating recommendations for the target user. The main part of this phase is the second task which aims to find the best-fit dynamic preference templates for recommendation generation. We employed a modified similarity measure proposed by Saneifar et al. (Saneifar et al, 2008) to find the best-fit dynamic preference templates. The modified similarity measure is a combination of weight of itemset mappings between two sequence patterns, and a mapping weight between two itemsets, itemset i of sequence Seq_1 and itemset j of sequence which are defined as:

$$Weight(i,j) = \frac{|Seq_1(i) \cap Seq_2(j)|}{(|Seq_1(i)| + |Seq_2(j)|)/2}$$

When computing the mapping weights, a selected itemset j from sequence Seq_2 corresponding to the itemset i of sequence Seq_1 may have already been mapped with another itemset p . This situation is called "mapping conflict", that we must find another itemset mapping to solve the conflict problem. Therefore, a local relevance evaluation *localSim* is adopted to evaluate possible mappings. The mapping pair with the highest local relevance is chosen to solve the conflict. The calculation of *localSim* would vary from different types of mappings:

$$localSim(i, c_1)(j, c_2) = \begin{cases} (weight(i, c_1) + weight(j, c_2))/2, & \text{when the mappings comply with order} \\ 1/2 \times (weight(j, c_1) + weight(i, c_2))/2, & \text{when the mappings are cross} \end{cases}$$

where c_1, c_2 are candidate mapping itemsets

After solving all the mapping conflicts, the similarity degree between two sequences is calculated with the following equation:

$$Sequence_Similarity = \frac{\sum_{q=1}^k weight_q}{\max(|Seq_1|, |Seq_2|)}$$

Let us take $Seq_1 = \{(bc)(df)(e)\}$ and $Seq_2 = \{(abc)(mn)(de)(egh)(fg)\}$ as an example to show the computation of similarity degree between the two sequences. For each itemset $Seq_1(i)$, we firstly look for the most similar itemset $Seq_2(j)$ in the second sequence Seq_2 . The best mapping for the 1st itemset (bc) in Seq_1 is itemset (abc) in Seq_2 , and the mapping weight is calculated as: $2/((3+2)/2) = 0.8$. Then, we continued to search the mapping with the 2nd itemset (df) in Seq_1 and found that (de) and (fg) are two mapping candidates of (df). After computing the second run mapping weight, we obtained $weight((df),(de)) = 1/((2+2)/2) = 0.5$ and $weight((df),(fg)) = 1/((2+2)/2) = 0.5$. Since the two mapping

weights are equal, we chose the itemset (de) with lower timestamp to map with the itemset (df) in Seq_1 .

We did another search for the 3rd itemset (e) in Seq_1 , and found that (de) in Seq_2 is the best mapping itemset with $weight((e),(de))=0.6$. However, this itemset (de) has already been associated with the (df) of Seq_1 . Thus a mapping conflict happens, then the conflict-solving procedure is activated to find new mapping pairs for itemsets (df) and (e). The possible mapping itemsets, which is located before or after the current mapped itemset (de) in Seq_2 , has become the new candidates by then. So (fg) and (egh) are the possible mapping candidates for (df) and (e). Consequently, the new possible mappings for the conflict are: $\langle((df),(de)); ((e),(egh))\rangle$ and $\langle((df),(fg)); ((e),(de))\rangle$. Using the local relevance evaluation and considering case of cross-mapping, the calculations of $localSim$ are as followings:

$$localSim((df),(de))((e),(egh))=(0.5+0.5)/2=0.5$$

$$localSim((df),(fg))((e),(de))=1/2 \times (0.6+0.5)/2=0.27$$

The mapping pairs $\langle((df),(de)); ((e),(egh))\rangle$ with the highest $localSim$ are selected to solve the conflict and the new mapping of itemset (e) is (egh). The final itemset mappings between the two sequences are: (abc),(ab); (df),(de); (e),(egh). Finally, the estimation of the similarity degree laid between Seq_1 and Seq_2 is: $(0.8+0.5+0.5)/\max(3,5)=0.36$.

After selecting the best-fit dynamic preference profiles for the target user, the recommendation content is generated according to the selected dynamic preference profile with the Top- k method.

4 EMPIRICAL EVALUATION

With taking user interest evolution patterns into account and implementing the traditional collaborative filtering approach as the performance benchmark, we conduct an empirical evaluation for the proposed recommendation approach. First, the evaluation dataset is depicted in Section 4.1. Then, the evaluation procedure and performance criteria are presented in Section 4.2. Next, the tuning experiments on the effects of related parameters for the proposed 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 and description

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. Averagely, each subject offered 106 preference records. Because the MovieLens dataset does not contain the description of each movie, we adopted the plot keywords from IMDB movie database to represent each movie as a feature (keyword) vector accordingly.

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 possible bias, the experiments in section 4.3 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. Besides, the traditional collaborative recommendation was adopted as the comparative recommendation technique. Meanwhile, the Top k method was applied to generate recommendation results.

Furthermore, we adopt precision as the performance evaluation criteria to evaluate the prediction accuracy of our proposed approach (namely SPIE) and the traditional collaborative filtering approach

(namely CF) for the recommendation results. The precision criterion is a widely adopted measure to evaluate the prediction accuracy according to intersection percentage between the actual user’s favor items and the recommended items. The calculation is defined as:

$$\frac{|Actual_Preference \cap Recommendations|}{|Recommendations|}$$

4.3 Parameters Tuning Results

In the designed 30 evaluation experiments, each $D_{training}$ averagely includes 582 users and each $D_{testing}$ averagely includes 500 users. There are averagely 139 users which appear in both $D_{training}$ and $D_{testing}$ in each experiment, thus the 139 users, intersection of $D_{training}$ and $D_{testing}$, are employed to evaluate the performance of personalized recommendation in each experiment.

This study adopted the k-means clustering method to create item clusters and the number of item clusters will range from 4 to 15. The analysis of $D_{testing}$ in 30 experiments shows that each user included in the $D_{testing}$ had rated 2.00 to 6.47 item clusters based on the mentioned 4 to 15 item-clusters as shown in Table 1.

Item clusters	4	5	6	7	8	9	10	11	12	13	14	15
Rated clusters of each user in $D_{testing}$	2.00	2.17	2.70	3.17	3.69	4.11	4.31	5.06	5.28	5.83	6.47	6.47

Table 1. Average number of rated item clusters for each user

We also examined the effects of the support setting of GSP on number of mined sequential patterns, ranging from 0.9 to 0.1 at decrement of 0.1 by given the number of item clusters as 4, 5, 6, and 10. As shown in Table 2, the number of sequential patterns (i.e., users’ dynamic profile) has a positive relationship with the number of item clusters and a negative relationship with the support setting. For example, when the number of item cluster is fixed to 4, 5 or 6, the support should be lowered (e.g., 0.2) in order to acquire more sequential patterns (e.g. at least 10 patterns). Accordingly, the support rate of GSP is set as 0.2 in the following experiments in order to obtain sufficient dynamic profiles.

Support \ Cluster	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
4	1	3	3	3	3	3	3	10
5	1	3	3	3	3	5	7	10
6	3	3	7	7	7	7	7	20
10	6	8	15	23	31	43	63	95

Table 2. Effects of support on number of mined sequential patterns

When the support is set as 0.2 for GSP, there are 10 to 903 dynamic preference profiles extracted according to various item cluster settings as shown in Table 3. We then examined the effects of selected templates on recommendation precision performance, ranging 4 to 10 at increment of 2. As shown in Table 4, when item cluster settings are larger than 7, the proposed approach SPIE could achieve about 70% prediction accuracy while the number of selected templates is set as 10. Taking the richness of recommendation content and recommendation performance into account, we set the number of selected templates as 10 for the following comparative evaluations.

Item cluster	4	5	6	7	8	9	10	11	12	13	14	15
Extracted Profiles	10	10	20	28	54	95	98	183	279	470	888	903

Table 3. Number of mined dynamic profiles with various cluster settings

Number of profiles \ Item cluster	4	6	8	10
4	64.97%	53.61%	52.92%	52.92%
5	66.19%	56.26%	55.34%	55.34%
6	67.82%	66.71%	65.79%	65.62%
7	70.00%	69.06%	67.96%	67.89%
8	74.14%	74.12%	73.85%	73.86%
9	71.75%	71.69%	71.56%	71.56%
10	70.70%	70.65%	70.45%	70.45%
11	68.53%	68.50%	68.46%	68.46%
12	69.58%	69.57%	69.54%	69.54%
13	70.95%	70.94%	70.93%	70.93%
14	70.28%	70.27%	70.27%	70.27%
15	70.34%	70.35%	70.35%	70.35%

Table 4. Recommendation performance with various templates settings

4.4 Comparative Evaluation Results

To evaluate the performances of the proposed approach (SPIE), we use the traditional collaborative filtering approach (CF), a widely used recommendation technique, as the performance benchmark. We set support of GSP as 0.2 and the number of selected templates for recommendation as 10 for SPIE. For CF, we set the number of neighbors as 3 according to the parameter tuning experience on the same dataset (i.e., MovieLens) based on a recommendation research related to collaborative filtering (Cheng et al., 2011). Meanwhile, both techniques employ Top- k ($k=3$) strategy to generate recommendation contents. The comparative evaluation results are shown in Table 5.

We adopted a two-tailed T test to assess the diversity of two techniques on recommendation performance, the unveiled result is that all the comparison results of p -value are less than 0.01 and close to 0. The statistical results show that SPIE significantly outperforms CF on recommendation precision on various item cluster settings. In summary, the results have strongly suggested that our proposed approach does improve recommendation accuracy with users' interest change pattern. Besides, with this approach, recommendation would not be easily misguided simply because the basis of some preferences at a latest timestamp as traditional collaborative filtering approach does. Moreover, our approach also considers the item heterogeneities such as the preferences on the irrelevant items (i.e., the items with dissimilar contents) will not be utilized. As a result, our proposed approach avoids utilizing unreliable preferences which might have decreased the recommendation accuracy.

Technique \ Item cluster	SPIE	CF
4	52.92%	50.33%
5	55.34%	48.14%
6	65.62%	55.15%
7	67.89%	58.41%
8	73.86%	62.71%

9	71.56%	63.46%
10	70.45%	62.16%
11	68.46%	64.71%
12	69.54%	64.84%
13	70.93%	64.41%
14	70.27%	61.80%
15	70.35%	62.36%

Table 5 Comparative evaluation results of SPIE and CF

5 CONCLUSION

Collaborative recommendation is one of the most successful and most widely adopted recommendation techniques. The traditional collaborative recommendation technique assumes that customers' preferences obtained at different timestamps would have had the same value of importance. However, the fact that customers preference might change over time has not been in its consideration. Consequently, traditional collaborative technique fails to fully utilize the changing information of customers' purchase interest that is used in predicting customers' preference items. Therefore, we proposed the SPIE technique based on users' interest evolution patterns. The SPIE can extract the customer's dynamic preference profile from their historical purchase records to describe how their preferences change at different timestamps. Meanwhile, the SPIE also uses item clustering method to solve the sparsity problem on traditional collaborative recommendation technique. In this study, we adopted MovieLens datasets to implement empirical evaluation. The results of the empirical evaluation showed that the application of users' interest evolution patterns could significantly improve the recommendation process. Thus, SPIE's recommendation performance is better than CF's in precision. The result of this research also verified the assumption that the use of changes in customers' preferences will enhance the performance of recommendation system.

ACKNOWLEDGEMENT

This work was supported by the National Science Council of the Republic of China under the grant NSC 99-2410-H-218-022-MY2.

References

- Adomavicius, G. and Tuzhilin, A. (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), pp.734-749.
- Agrawal, R. and Srikant, R. (1995). Mining Sequential Patterns. *Proceedings of the 11th International Conference Data Engineering*, pp.3-14.
- Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based, Collaborative Recommendation. *Communications of the ACM* (40:3), pp.66-72.
- Bettini, C., Wang, X.S. and Jajodia, S. (1998). Mining Temporal Relationships with Multiple Granularities in Time Sequences. *Data Engineering Bulletin* (21), pp.2-38.
- Beyah, G., Xu, P., Woo, H.G., Mohan, K. and Straub, D. (2003). Development of an Instrument to Study the Use of Recommender Systems. *Proceedings of the Ninth Americas Conference on Information Systems*, pp.269-279.
- Cheng, T.H., Chen, H.C., Lin, W.B., and Lee, Y.H. (2011). Collaborative Filtering with User Interest Evolution. *Proceedings of the 15th Pacific Asia Conference on Information Systems (PACIS), Brisbane, Australia, July 2011, (PACIS 2011)*.

- Cho, Y.B., Cho, Y.H. and Kim, S.H. (2005). Mining Changes in Customer Buying Behavior for Collaborative Recommendations. *Expert System with Applications* (28:2), pp.359-369.
- Ding, Y. and Li, X. (2005). Time Weight Collaborative Filtering. *Processings of Information and Knowledge Management*.
- Ding, Y., Li, X. and Orłowska, M. E. (2006). Recency-Based Collaborative Filtering. *Proceedings of the 17th Australasian Database Conference*, pp.99-107.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations* (11:1), pp.10-18.
- Häubl, G. and Trifts, V. (2000). Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids. *Marketing Science* (19:1), pp.4-21.
- Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J.T. (1999). An Algorithmic Framework for Performing Collaborative Filtering. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.230-237.
- Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* (22:1), pp.5-53.
- Hijikata, Y., Shimizu, T. and Nishida, S. (2009). Discovery-Oriented Collaborative Filtering for Improving User Satisfaction. *Proceedings of the 13th international Conference on intelligent User interfaces*, pp.67-76.
- Khatri, M. (2012). A Study of Collaborative Filtering Approach for Temporal Dynamic Web Data. *Int. J. Advance Networking and Applications* (4:2), pp.1568-1573.
- Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. and Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM* (40:3), pp.77-87.
- Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *Proceedings of the 12th International Conference on Machine Learning*, pp.331-339.
- Larsen, B. and Aone, C. (1999). Fast and Effective Text Mining Using Linear-time Document Clustering. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.16-22.
- Mannila, H., Toivonen, H. and Verkamo, A.I. (1995). Discovering Frequent Episodes in Sequences. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pp.210-215.
- Min, S.H. and Han, I. (2005). Detection of The Customer Time-Variant Pattern for Improving Recommender Systems. *Expert Systems with Applications* (28:2), pp.189-199.
- Mobasher, B., Dai, H., Luo, T. and Nakagawa, M. (2001). Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. *Proceedings Of the Workshop Intelligent Techniques for Web Personalization*, pp.53-60.
- Mooney, R.J. and Roy, L. (1999). Content-based Book Recommending using Learning for Text Categorization. *Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation*.
- Pazzani, M. and Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning* (27), pp.313-331.
- Pei, J., Han, J., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.C. (2001). PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth. *Proceedings of International Conference on Data Engineering*, pp.215-224.
- Rolland, P. (2001). FIEExPat: Flexible Extraction of Sequential Patterns. *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp.481-488.
- Roussinov, D. G. and Chen, H. (1999). Document Clustering for Electronic Meetings: An Experimental Comparison of Two Techniques. *Decision Support Systems* (27:1-2), pp.67-79.
- Saneifar, H., Bringay, S., Laurent, A. and Teisseire, M. (2008). S2MP: Similarity Measure for Sequential Patterns. *Proceedings of The Australasian Data Mining Conference*.
- Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp.158-167.
- Schafer, J.B., Konstan, J.A. and Riedl, J. (2001). E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery* (5:1), pp.115-153.

- Srikant, R. and Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. *Proceedings of the 5th International Conference on Extending Database Technology*, pp.3-17.
- Strasser, M. and Zugenmaier, A. (2003). Personalization Through Mask Marketing. *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)*.
- Symeonidis, P., Nanopoulos, A., Papadopoulos, A.N. and Manolopoulos, Y. (2008). Collaborative Recommender Systems: Combining Effectiveness and Efficiency. *Expert Systems with Applications* (34:4), pp.2995-3013.
- Tang, T.Y., Winoto, P. and Chan, K.C. (2003). On the Temporal Analysis for Improved Hybrid Recommendations. *Proceedings of IEEE International Conference on Web Intelligence*, pp.214-220.
- Terveen, L., McMackin, J., Amento, B. and Hill, W. (2002). Specifying Preferences Based on User History. *Proceedings of Conference on Human Factors in Computing Systems*, pp.315-322.
- Wan, Y., Menon, S. and Ramaprasad, A. (2003). How IT Happens: A Conceptual Explanation of Choice Overload in Online Decision Making by Individuals. *Proceedings of Ninth Americas Conference on Information Systems*, pp.2379-2384.
- Wei, C., Sen, W., Yuan, Z. and Chen, L.C. (2009). Algorithm of Mining Sequential Patterns for Web Personalization Services. *SIGMIS Database* (40:2), pp.57-66.
- Wei, C., Shaw, M.J. and Easley, R.F. (2002). A Survey of Recommendation Systems in Electronic Commerce. *E-Service: New Directions in Theory and Practice*. R. T. Rust and P. K. Kannan (Eds), M. E. Sharpe Publisher.
- Wei, Y.Z., Moreau, L. and Jennings, N.R. (2005). A Market-Based Approach to Recommender Systems. *ACM Transactions on Information Systems* (23:3), pp.227-266.
- Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning* (42:1-2), pp.31-60.
- Zhao, Q. and Bhowmick, S. (2003). Sequential Pattern Mining: A Survey. *Technical report*, Nanyang Technichal University, Singapore.
- Zhao, Y., Zhang, C. and Zhang, S. (2005). A Recent-Biased Dimension Reduction Technique for Time Series Data. *In Advances in Knowledge Discovery and Data Mining: Lecture Notes in Computer Science*, pp.107-176.