Personalized Recommendation Based on Weighted Sequence Similarity

Wei Song and Kai Yang

Abstract The sequential pattern mining-based recommendation has recently become a popular research topic in the field of recommender system. However, this kind of methods usually relies on frequency counts of sequences, which makes low-frequency sequences contribute little for the final recommend results. To solve this problem, in this paper, we propose a weighted sequence similarity-based method, called Personalized Recommendation based on Sequence Similarity (PRSS), for personalized recommendation. First, item-sequence weight model is introduced, which can reflect different importance of different items to different sequences. Then, target users' sequence is compared with historical sequences using similarity function. Finally, the maximal common subsequence is proposed to rank candidate sequences and make recommendation. Experimental results show that PRSS generates more accurate recommendation for the target users.

Keywords Recommender systems · Sequential pattern · Item-sequence weight · Similarity function · Maximal common subsequence

1 Introduction

A recommender system is a Web technology that suggests items of interest to users based on their objective behavior or their explicitly stated preferences [7]. Such systems typically provide the user with a list of recommended items they might prefer, or supply guesses of how much the user might prefer each item. These systems help users to decide on appropriate items, and ease the task of finding interesting items in the collection.

W. Song (✉) · K. Yang
College of Information Engineering, North China University of Technology,
Beijing 100144, China
e-mail: songwei@ncut.edu.cn

Z. Wen and T. Li (eds.), *Practical Applications of Intelligent Systems*,
Advances in Intelligent Systems and Computing 279,
DOI: 10.1007/978-3-642-54927-4_62, © Springer-Verlag Berlin Heidelberg 2014

Content-based recommender systems try to recommend items similar to those a given user has liked in the past [1, 8]. Indeed, content-based recommender systems analyze item descriptions to identify items that are particular preferred by the user. User preferences can be captured using four types of data: demographic data, rating data, browsing pattern data, and transaction data. Transaction data provides sets of preferred items and can be used to predict future customer preferences. Therefore, some researchers applied the association rule mining technique to extract the accessing sequences to improve recommender system performance [4, 6]. However, such systems incorporate customer transaction data from only a single temporal period, which omits the dynamic nature of a customer's access sequences.

Unlike association rules, sequential patterns [2] may suggest that a user who accesses a new item in the current time period is likely to access another item in the next time period. Thus, sequential pattern mining techniques have been used for recommendation recently [3, 10]. However, there are two main drawbacks of the existing recommendation methods based on sequential pattern. On the one hand, only high-support sequential patterns are used for recommendation, but the simple frequency counts of patterns are not always effective. On the other hand, these methods usually treat all sequences as equally important. However, in real applications, sequential patterns often carry varying significance with respect to each target user.

In this paper, we study the problem of personalized recommendation from the perspective of different effects of different items to different sequences. To realize this target, the item-sequence weight model is introduced at first. Then, by representing sequences as vectors, different similarity functions are discussed. To reflect the alignment similarity of sequences, maximal common subsequence is proposed to rank candidate sequences and make recommendation. Experimental results show that the proposed Personalized Recommendation based on Sequence Similarity (PRSS) method outperforms related approach in terms of accuracy.

2 Related Work

Association rule is a widely used data mining technique that generates recommendations in recommender systems [4, 6]. More specifically, the method tries to discover the relationships between product items based on patterns of co-occurrence across customer transactions. These association rule-based methods are comprised of three steps: (1) Find an association between two sets of products in a transaction database; (2) The active customer's purchase behavior is compared with the discovered association rule base to find the most similar purchasing behaviors; (3) A set of products that the active customer is most likely to purchase is then generated by selecting the top-N most commonly purchased products. However, association rule mining does not take the time stamp into account. Thus, the association rule-based recommender system cannot reflect the dynamic nature of users' behavior.

Unlike association rules, sequential patterns [2] may suggest that a consumer who buys a new product in the current time period is likely to buy another product in the next time period. While association rule discovery covers only intra-transaction patterns (itemsets), sequential pattern mining also discovers inter-transaction patterns (sequences). In essence, frequent sequential pattern discovery can be thought of as association rule discovery over a temporal database. As association rules can be used for recommendation, sequential pattern mining-based recommendation algorithms have been studied for online product recommendation in recent years [10].

The sequential pattern mining-based recommendation is to induce a predictive model from a set of examples (i.e., the sequences in database). The usefulness of sequential pattern mining-based recommendation has, to a certain extent, been demonstrated empirically by past studies on various domains such as Web browsing [10], e-commerce [3], and music [5]. However, a significant shortcoming is that these methods do not perform user-specific sequential pattern mining and, as a result, they cannot give accurate personalized recommendation to users.

Recently, Yap et al. [9] proposed a personalized sequential pattern mining-based recommendation framework. A competence score measure, considering relevance to the target user, and the sequences' recommendation ability, is used for accurate personalized recommendation. Furthermore, additional sequence knowledge is exploited to improve the efficiency and the quality of learning in sequential pattern mining algorithms. However, their framework still depends on support threshold. Thus, if some sequences do not have enough supporting samples, they cannot be used for recommendation.

3 Problem Definition

Let $\boldsymbol{I} = \{i_1, i_2,\ldots, i_m\}$ be a finite set of *items*, set $X \subseteq \boldsymbol{I}$ is called an *itemset*. A *sequence* α is denoted by $<X_1, X_2,\ldots, X_m>$, where X_j $(1 \le j \le m)$ is an itemset. X_j is also called an *element* of the sequence, and denoted as $(x_1, x_2,\ldots, x_n)$, where x_k $(1 \le k \le n)$ is an item. Each itemset X_i comprises items with the same timestamp $t(X_i)$, and the different itemsets X_i and X_j in the same sequence cannot have the same timestamp, i.e., $t(X_i) \ne t(X_j)$ $(i \ne j)$. The number of instances of items in a sequence is called the length of the sequence.

For example, a sequence <(1, 4) (3) (2, 8) (1, 5)> describes a user who accessed the items 1 and 4 at timestamp t_1, item 3 at timestamp t_2, items 2 and 8 at timestamp t_3, and items 1 and 5 at timestamp t_4.

A sequence database ***SDB*** is a collection of sequences, i.e., $\boldsymbol{SDB} = \{S_1, S_2,\ldots, S_n\}$ where $|\boldsymbol{SDB}| = n$ denotes the number of sequences (also the number of users as each user has a corresponding sequence in ***SDB***). A sequence $S_i \in \boldsymbol{SDB}$ $(i = 1, 2,\ldots, n)$ is a ordered list of itemsets associated with a user U_i.

Given a target user's past sequence S_q and the database of all other users' past sequences, the personalized sequence-based recommendation task is to predict items that the target user is most likely to access in the near future, i.e., those items in the next few item sets that he or she will access.

4 Personalized Sequence-Based Recommendation

4.1 The Item-Sequence Weight Model

In real applications, sequences often carry different significance with respect to each target user. To mine the personalized sequential patterns for a target user, we have to effectively model this varying relevance among the historical sequences for that specific user. Since each sequence in ***SDB*** belongs to a different user, we can set weights of sequences based on available knowledge on how relevant is each of the sequences compared to the known sequence of the target user. We solve this problem by considering items' appearance in different sequences.

Definition 1 Given sequence database ***SDB*** and an item b, $SS_b = \{S_i \mid b \in S_i \in \boldsymbol{SDB}\}$ is the set of sequences containing item b, then the *item-sequence weight* of item b with respect to sequence S_i is defined as:

$$w_{S_i}(b) = \frac{T_{S_i}(b)}{L(S_i)} \times \frac{|\mathrm{SDB}|}{|SS_b|} \tag{1}$$

where $T_{S_i}(\alpha)$ is the number of occurrences of b in S_i; $L(S_i)$ is the length of sequence S_i, i.e., number of total items in sequence S_i; $|\boldsymbol{SDB}|$ and $|SS_b|$ are number of sequences in ***SDB*** and SS_b, respectively.

The above formula of item-sequence weight consists of two components. The one before sign of multiplication is measured for item b in a sequence S_i. On the contrary, the one after sign of multiplication shows the influence degree of the sequence containing item b within the whole database.

In this paper, the item-sequence weight of each pair of item and sequence is calculated and is used to measure whether an item occurs many times in certain sequences but with less influence to other sequences. The items with high item-sequence weight are considered with high probabilities of recommendation for sequences containing them.

4.2 Weighted Sequence Matching

Given item-sequence weight, we can transform each user's sequence S_i into a *sequence vector* $\overrightarrow{v_i}$, in which each dimension denotes the item-sequence weight of a unique item. That is, a sequence vector $\overrightarrow{v_i}$ is composed of the ordered item-sequence weights of items contained in sequence S_i.

Given a target user's sequence S_q and its sequence vector $\overrightarrow{v_q}$, for each S_i in sequence database ***SDB*** and its sequence vector $\overrightarrow{v_i}$, we can calculate the similarity between S_q and S_i using the following two functions.

1. Distance-based similarity.

$$\mathrm{sim}\left(S_q, S_i\right) = \frac{1}{1 + \sqrt{\sum_{k=1}^{m} \left(x_{qk} - x_{ik}\right)^2}} \tag{2}$$

where x_{qk} and x_{ik} are the kth dimension of sequence vectors $\overrightarrow{v_q}$ and $\overrightarrow{v_i}$, respectively.

2. Cosine similarity.

$$\mathrm{sim}\left(S_q, S_i\right) = \frac{\overrightarrow{v_q} \cdot \overrightarrow{v_i}}{|\overrightarrow{v_q}||\overrightarrow{v_i}|} \tag{3}$$

Given a similarity threshold δ ($0 \leq \delta \leq 1$), only sequences with similarity no lower than δ are used for recommendation.

4.3 Computing Sequences for Recommendation

The two similarity measures discussed in Sect. 4.2 tend to reflect the content similarity rather than the alignment similarity. To compute a personalized score for each sequence S_i in the sequence database to reflect its competency in recommendation for the target user, we propose the following maximal common subsequence to rank the candidate sequences selected by distance-based similarity or cosine similarity.

Definition 2 Given a target user's sequence S_q and sequence S_i in sequence database ***SDB***, if there are n common subsequences of S_q and S_i, the *maximal common subsequence* (MCS) of S_q and S_i is defined as:

$$\mathrm{MCS}\left(S_q, S_i\right) = \frac{\max(L(\mathrm{CS}_1), \ldots, L(\mathrm{CS}_n))}{L(S_q)} \tag{4}$$

where CS_i ($1 \leq i \leq n$) is the ith common subsequences of S_q and S_i, and $L(\mathrm{CS}_i)$ is the length of sequence CS_i.

Compared with the two similarity measures discussed in Sect. 4.2, maximal common subsequence is not only compatible to the user sequence S_q, but also is capable of readily extending beyond S_q to offer more next-items. In this paper, within sequences selected by the similarity function, only the one with highest maximal common subsequence value is used for recommendation.

4.4 Algorithm Description

Based on the above discussion, the proposed PRSS is described in Algorithm 1.

```
Algorithm 1   PRSS
Input         The target user's sequence $S_q$, $\boldsymbol{SDB}$ (a sequence database),
              similarity threshold $\delta$
Output        The next item recommended to the target user
1)    Compute the sequence vector of the target user 's sequence $S_q$;
2)    for each sequence $S_i$ in $\boldsymbol{SDB}$ do
3)        Compute the sequence vector of $S_i$.
4)        if $sim(S_q, S_i) < \delta$ do
5)            continue;
6)        else
7)            $S_i \rightarrow CS_q$;
8)            Compute $MCS(S_q, S_i)$ using Eq. (4);
9)        end else
10)       end if
11)   end for
12)   Recommend the next item using the sequence in $CS_q$ with highest MCS value.
```

In Algorithm 1, item-sequence weights of items of the target user are calculated using Eq. (1) at first. Then, the target user's sequence is represented by vector composed of item-sequence weights. The similarities between target user sequence and sequences in ***SDB*** are computed in the main loop (steps 2–11). Those sequences with similarities higher than threshold are kept in the set CS_q (step 7), and the MCS values of them with the target user's sequence are recorded (step 8). Step 12 recommends to the target user with the sequence with maximal MCS value.

5 Experimental Evaluation

We present a two-part evaluation of our proposed PRSS. In Sect. 5.2, we compare the effectiveness of using different similarity measurements. In Sect. 5.3, we compare the effectiveness in terms of F1-measure of PRSS with the recommendation method (denoted by NRPS in this paper) proposed in [9].

The experiments were performed on a Pentium Dual E2140 1.60 GHz CPU with 2 GB memory, and running on Windows XP. We use the msnbc.com dataset from UCI (http://www.ics.uci.edu/~mlearn) for the performance evaluation. It captures the time-ordered sequence of Web pages visited by msnbc users on a day. There are a total of 989,818 sequences, each one corresponding to a different user. To evaluate the recommendation effect on sequences with different lengths, we divide the dataset into seven subsets: I: sequences with lengths in domain of (0, 35]; II: sequences with lengths in domain of (35, 45]; III: sequences with lengths in domain of (45, 55]; IV: sequences with lengths in domain of (55, 65]; V: sequences with lengths in domain of (65, 75]; VI: sequences with lengths in domain of (75, 85]; VII: sequences with lengths in domain of (85, 95].

5.1 Evaluation Measures

The F1-measure performance measure is considered in evaluating the effectiveness of the proposed method. The F1-measure measure is based on two performance measures precision and recall.

Precision is the fraction of recommended product items that are considered interesting, as defined in Eq. (5).

$$\text{precision} = \frac{\text{Number of correctly recommended items}}{\text{Number of recommended items}} \tag{5}$$

In contrast, *recall* is the ratio that the successfully recommended items to all of the accessed items, as defined in Eq. (6).

$$\text{recall} = \frac{\text{Number of correctly recommended items}}{\text{Number of accessed items}} \tag{6}$$

Precision measures how many of the recommended items belong to the actual customer access list, whereas recall measures how many of the items in the actual customer access list consist of recommended items. These measures are simple to compute and intuitively appealing, but they are in conflict, since increasing the size of the recommendation set leads to a decrease in precision but, at the same time, to an increase in recall. Hence, a widely used combination F1-measure Eq. (7), which gives equal weight to both recall and precision, was also employed in the course of our evaluation.

$$\text{F1-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

5.2 Comparison on Different Similarity Functions

The objective of this experiment is to find a similarity function appropriate to the msnbc.com dataset. To do this, we used the two similarity functions (i.e., distance-based similarity, and cosine similarity) mentioned in Sect. 4.2, and compared the F1-measure. The comparison results are shown in Fig. 1.

For distance-based similarity, the trend of F1-measure is not steady, with the highest value 0.828 and the lowest value 0.381. For cosine similarity, F1-measure increases with the increases of sequences' lengths when sequences' lengths are no higher than 55, while it decreases with the increases of sequences' length when the sequences' lengths are in domain of (55, 75]. The best F1-measure was obtained when sequences' lengths are longer than 75. From the results, we can conclude that cosine similarity is more appropriate for msnbc.com dataset.

5.3 Accuracy Comparison

In this subsection, we compare the performance of the proposed PRSS with NRPS [9] with regard to F1-measure. As NRPS use high-support sequential patterns to recommend, in this set of experiments, minimum support threshold (denoted by min_sup) is also used besides the similarity threshold.

Figure 2 shows the F1-measure comparisons when $\delta = 90\ \%$ and min_sup = 10 %. Although the F1-measure of PRSS is lower than that of NRPS on dataset with sequences lengths no longer than 35, on average, the F1-measure of PRSS is 43 % higher than that of NRPS.

Figure 3 shows the F1-measure comparisons when $\delta = 92\ \%$ and min_sup = 30 %. The result is almost the same as shown in Fig. 2. On average, the F1-measure of PRSS is 49 % higher than that of NRPS.

6 Conclusions

We propose a personalized recommendation method by considering effects of different items for different sequences. Specifically, item-sequence weight model is introduced for representing importance of items and sequences at first. Then, different similarity functions are discussed for selecting appropriate sequences for recommendation. Finally, the maximal common subsequence is proposed for ranking candidate sequences and making recommendation. Compared with related sequence-based recommendation methods, the proposed PRSS method does not rely on support. Thus, some low-frequency important sequences will not be ignored. Experimental results show that PRSS method can achieve more accurate recommendation results.

Fig. 1 Comparison of F1-measure using different similarity function, $\delta = 90\ \%$

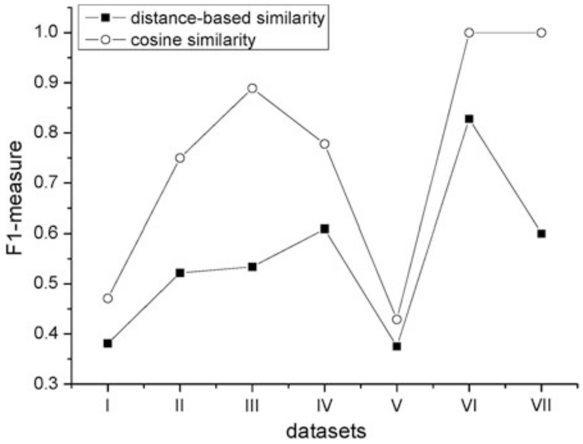


Fig. 2 Comparison of F1-measure between two approaches, $\delta = 90\ \%$, min_sup = 10 %

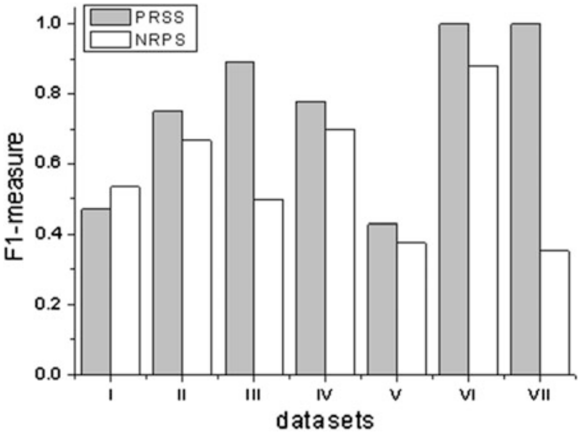
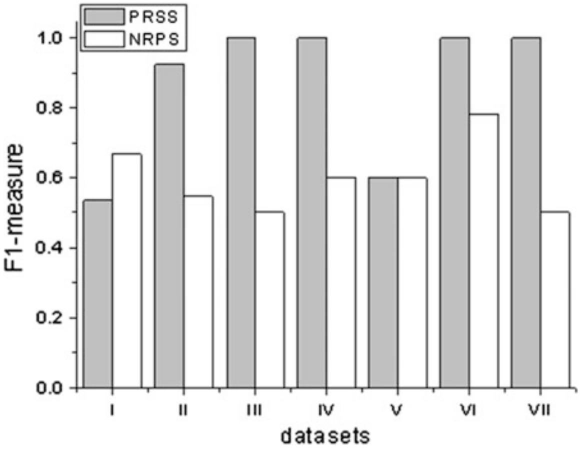


Fig. 3 Comparison of F1 between two approaches, $\delta = 92\ \%$, min_sup = 30 %



Acknowledgments The work is partly supported by the National Natural Science Foundation of China (61105045), Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality (PHR201108057), Funding Project for Academic Human Resources Development of North China University of Technology (CCXZ201303), and key project of North China University of Technology.

References

1. Agarwal D, Chen B-C, Elango P, Ramakrishnan R (2013) Content recommendation on web portals. Commun ACM 56:92–101
2. Febrer-Hernández JK, Palancar JH (2012) Sequential pattern mining algorithms review. Intell Data Anal 16:451–466
3. Huang C-L, Huang W-L (2009) Handling sequential pattern decay: developing a two-stage collaborative recommender system. Electron Commer R A 8:117–129
4. Kazienko P, Pilarczyk M (2008) Hyperlink recommendation based on positive and negative association rules. New Generation Comput 26:227–244
5. Liu N-H (2013) Comparison of content-based music recommendation using different distance estimation methods. Appl Intell 38:160–174
6. Paranjape-Voditel P, Deshpande U (2013) A stock market portfolio recommender system based on association rule mining. Appl Soft Comput 13:1055–1063
7. Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. Expert Syst Appl 39:10059–10072
8. Pera MS, Ng Y-K (2013) A group recommender for movies based on content similarity and popularity. Inf Process Manage 49:673–687
9. Yap G-E, Li X-L, Yu PS (2012) Effective next-items recommendation via personalized sequential pattern mining. In: Proceedings of 17th international conference on database systems for advanced applications, pp 48–64
10. Zhou B, Hui SC, Fong ACM (2006) Efficient sequential access pattern mining for Web recommendations. KES J 10:155–168