Collaborative Filtering Recommendation on Users’ Interest Sequences

Weijie Cheng *, Guisheng Yin, Yuxin Dong, Hongbin Dong, Wansong Zhang

College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, China

* chengweijie@hrbeu.edu.cn

Abstract

As an important factor for improving recommendations, time information has been introduced to model users’ dynamic preferences in many papers. However, the sequence of users’ behaviour is rarely studied in recommender systems. Due to the users’ unique behavior evolution patterns and personalized interest transitions among items, users’ similarity in sequential dimension should be introduced to further distinguish users’ preferences and interests. In this paper, we propose a new collaborative filtering recommendation method based on users’ interest sequences (IS) that rank users’ ratings or other online behaviors according to the timestamps when they occurred. This method extracts the semantics hidden in the interest sequences by the length of users’ longest common sub-IS (LCSIS) and the count of users’ total common sub-IS (ACSIS). Then, these semantics are utilized to obtain users’ IS-based similarities and, further, to refine the similarities acquired from traditional collaborative filtering approaches. With these updated similarities, transition characteristics and dynamic evolution patterns of users’ preferences are considered. Our new proposed method was compared with state-of-the-art time-aware collaborative filtering algorithms on datasets MovieLens, Flixster and Ciao. The experimental results validate that the proposed recommendation method is effective and outperforms several existing algorithms in the accuracy of rating prediction.

Introduction

The existing recommendation systems have adopted various methods to derive people’s preferences and interests. All such methods can be divided into three categories: content-based recommendation, collaborative filtering-based recommendation (CF) and hybrid recommendation. Among these three approaches, the collaborative filtering approach is one of the most successful. It requires only users’ past behavior, such as their item ratings, browsing history and purchased items, without requiring more extensive knowledge. Over the past decade, neighbor-based CF and latent factor model-based CF approaches have been proposed, and their effectiveness and efficiency have been verified in recommendation systems. In order to improve the CF approaches, many efforts have been carried out. Zike Zhang et al. utilized social tags to solve well-known cold-start problem in social tagging systems [1]. A scaling-based algorithm with
tunable parameters was introduced to promote personalized recommendation in solving the accuracy-diversity dilemma, presenting a high novelty and solving cold-start problem [2]. Based on classical Matrix Factorization (MF), multiple possible pairwise relationship affecting the final rating decisions of online users was extracted and then linearly integrated to predict their ratings [3]. In recent years, time information has been introduced into the CF approaches to model people’s dynamic evolving preferences and interests. Although contradictory conclusions can be drawn from the existing literature (some propose that the weight of more recent items should be increased, while others suggest that older items should not be underweighted), many researchers agree to some extent that users’ preference and interests change constantly over time and that time information indeed influences the performance of recommendation systems. In particular, the championship team from the Netflix Prize competition claimed that time information is crucial to their method [4][5].

However, the existing time-aware collaborative filtering approaches considered the time factor to be only an adaptive weight for adjusting users’ similarities [5][6][7] or rating history [8] directly; they ignored the sequence of the rated items. In our research we find that the sequence is non-trivial for improving recommendation accuracy because correlations and inter-item dependencies can cause various transition probabilities [9] among items for diverse users, and the differences between users’ characteristics and experiences can also lead to users’ unique rating sequences. In addition, sequences can reflect the evolutionary patterns of users’ preferences and experiences, which can to some extent reveal hidden information affecting their experiences and tastes. As every item has various features or functions that meet users’ tastes, the ratings on common items cannot expose the differences in users’ preference transition sequences, but the sequence of their ratings can. Furthermore, inspired from the analysis of temporal spatial sequences in location-based recommendation systems [10][11][12][13][14][15], this paper assumes that the interest sequences (IS) might also carry more semantics than single-interest points in on-line recommendation systems, and these semantics could be used to analyze the evolutionary patterns of users’ real dynamic interests. The assumption holds because users’ interests are not static but dynamic. These dynamic interest processes can be expressed using IS that contain interest transition sequences among items and the sequence of users’ behaviors. To analyze users’ dynamic interests, this paper gives a formal definition of “interest sequences” and designs methods to measure users’ similarities based on the longest common sub-IS (LCSIS) and all common sub-IS (ACSIS). A new similarity measurement method is designed by combining IS-based similarities with classic similarities from traditional CF methods. According to these new similarities, the other users in the system are ranked and then the top K users are selected as the target user’s neighbors. The rating prediction method that estimates users’ potential scores on unobserved items is presented using the neighbors’ observed ratings and the similarities among users.

The main contributions of this paper are as follows:

1. To depict users’ dynamic interest evolution patterns, we define the term “interest sequences” and other related concepts in on-line recommendation systems, which are inspired by work in location-based recommendation systems.

2. Based on the interest sequences, we introduce methods to calculate the length of LCSIS and the count of ACSIS, which are extended by taking into account the sequence and the deviations of users’ ratings on the common items.

3. Given the length of LCSIS and the count of ACSIS, we design the method to calculate users’ similarities based on IS, which are used to find the target user’s top K nearest neighbors. Finally, we present the method to predict users’ ratings for unrated items.
4. We adopt three datasets (MovieLens [16], Flixter [17] and Ciao [18]) to verify the effectiveness of the proposed recommendation method through comprehensive experiments and discuss the influence of several factors on the results.

Related Work

This paper proposes a new CF recommendation method based on users’ IS. In particular, we employ the sequential semantics contained in the interest sequences to select the most similar neighbors to predict a user’s preferences to items, so the proper items can be recommended. In this section, we review the most related research from the following two areas: CF recommendation approach and similarity measurement of time series data.

CF Recommendation Approach

The CF recommendation approach is one of the most successful and widely used recommendation techniques. CF makes predictions about a user’s ratings by collecting interests from many similar users [19]. The philosophy behind CF relies on the assumption that users who have had similar interests in the past are likely to share common interests in the future [20]. For decades, CF has been developed in two main categories: neighbor-based CF and latent factor model-based CF. Although recently, latent factor model-based CF using matrix factorization has obtained some of the best results [21][22][23][24], neighbor-based CF is still quite attractive due to its easy implementation. Neighbor-based CF relies on the similarities among users to select the most similar users as the target user’s neighbors and then predicts the ratings on the unobserved items using the neighbors’ observed ratings. For neighbor-based CF, similarity measurement is a critical design decision, and several different similarity functions have been proposed and evaluated in the literature. The most commonly used similarity functions include the Pearson Correlation Coefficient, Vector Space Similarity and commonly rated items between users [25]. The conventional neighbor-based CF has the great advantage of being able to integrate rich side information from users and items to refine similarities [26]. The most commonly used side information includes timestamps, which record the precise moments at which users interact with an item. As important context information, timestamps can be used to model users’ dynamic interests [27][28] and to tackle the evolution of user preferences [5][8][29][30][31]. Time is usually introduced as an adjusting factor to increase or reduce the weight of a user’s most recent ratings, but the sequence of users’ rating behaviors has rarely been studied; therefore, that topic is the main focus of this paper.

Similarity Measurement of Time Series Data

Time series data abounds in real world problems [32], and the measurement of their similarity is crucial in various applications such as bioinformatics, web mining and text mining [33]. Due to the sequential and temporal characteristics of time series data, the lengths of the longest common subsequences (LCS) are most commonly used to measure the distance between two time series [33][34][35][36]. CF can use the longest common subsequence as an indication of the similarity relationship between sequences, but that measurement fails to consider common information in other shorter common subsequences [32]. Hui Wang [32] suggested that the common information contained in all the common subsequences (ACS) should also be considered. The count of all common subsequences was introduced to improve the measure of similarity between time series [37][38]. Although the similarity of time series data has been successfully applied in classification problems, it has not been widely used in recommendation systems. Rajhans Mishraa, Pradeep Kumarb and Bharat Bhasker [34] developed a novel system that considered the sequential information present in web navigation patterns to recommend
websites to users, and they validated the viability of using LCS-based similarity in recommender systems. However, a similarity technique that integrates LCS length and the ACS count has not yet been studied in recommendation systems.

**Problem Statement**

The recommendation method in this paper is based on users’ interest sequences, which consider the sequences of users’ behaviors according to the time of their interactions. To elucidate our method, we provide some notations and definitions here to formalize the recommendation tasks addressed in this paper.

The recommendation task can be considered as a utility function indicating the potential interest of an item for a user. Let Users \( = \{u_1, \ldots, u_n\} \) denote a set of users, Items \( = \{i_1, \ldots, i_m\} \) denote a set of items, and \( Ts = \{t_1, \ldots, t_n\} \) denote all the timestamps at which the users rate items. To describe the IS-based recommendation system, we first provide some definitions below.

**Definition 1** An interest point IP stands for a user’s rating on an item at a timestamp, which consists of the user, the item, the rating, and the timestamp, that is, \( IP = \{ip_{u,it} = (u, it, r_i, ts)\} \), where \( u \) is the user, \( it \) is the item, and \( ts \) is the rating given by the user \( u \) on the item \( it \) at the timestamp \( ts \).

**Definition 2** A interest sequence IS stands for a sequence of a user’s interest points according to their time sequence, that is, \( is_u = (ip_{u_1}^{t_1} \rightarrow ip_{u_2}^{t_2} \rightarrow \cdots \rightarrow ip_{u_{l-1}}^{t_{l-1}}) \) where \( t_1 < t_2 < \cdots < t_{l-1} \). As the time sequence matters more than the concrete timestamps in this paper, IS can be written for short as \( is_u = ((it_1^u, r_{t_1}^u) \rightarrow (it_2^u, r_{t_2}^u) \rightarrow \cdots \rightarrow (it_l^u, r_{t_l}^u)) \), where \( l \) is the index that represents an IP’s position in the time sequence.

Based on the above definitions, we can transform all the users’ rating histories into interest sequences. Different from existing methods that use users’ ratings on the common items, this paper utilizes users’ IS to analyze users’ unique preferences because IS carries more semantics than standalone ratings so that it can not only show people’s dynamic interests but also indicate their evolution patterns. To calculate similarities between users’ IS, we take into account the length of the longest common sub-IS and the count of all common sub-IS, which have been verified as effective in classification problems [33][39][40][41]. To achieve the above tasks, we provide some additional definitions as follows:

**Definition 3** Interest Sequence Match (ISM). Given a rating deviation constraint threshold \( \theta \) and two users’ sub-IS \( sis_u = (it_1^u, r_{t_1}^u) \rightarrow (it_2^u, r_{t_2}^u) \rightarrow \cdots \rightarrow (it_l^u, r_{t_l}^u) \) and \( sis_v = (it_1^v, r_{t_1}^v) \rightarrow (it_2^v, r_{t_2}^v) \rightarrow \cdots \rightarrow (it_l^v, r_{t_l}^v) \) from two interest sequences \( is_u \) and \( is_v \), respectively, these two sub-IS formulate a \( j \)-length IS match if and only if they satisfy the following two conditions:

1. \( \forall i \in [1, j], it_i^u = it_i^v \);
2. \( \forall i \in [1, j], f_{dev}(r_i^u, r_i^v) \leq \theta \).

In the above definition, \( f_{dev}(r_i^u, r_i^v) \) is a function to calculate the deviation between the ratings given by user \( u \) and user \( v \) for the same item. It can be defined as in Eq (1)

\[
    f_{dev}(r_i^u, r_i^v) = \frac{r_i^u - \min(r^u) - r_i^v - \min(r^v)}{\max(r^u) - \min(r^u) - \max(r^v) - \min(r^v)}
\]

Due to the diversity of different users’ rating scales (some tolerant users may give 100 percent of the full mark, but other, more fastidious users may give only 70 percent to their favorite items), all the users’ ratings should be normalized to the same scale \([0, 1]\). If their deviation is smaller than the deviation constraint threshold \( \theta \), the two users’ ratings can be considered
equal. A smaller $\theta$ means a stricter similarity constraint, but a too-strict similarity constraint will limit the effect of IS in the recommendation. Therefore, an appropriate $\theta$ should be optimized according to an application’s sensitivity to IS.

With the defined ISM, LCSIS and ACSIS can be defined as follows:

**Definition 4** Longest Common Sub-IS (LCSIS). A ISM between two interest sequences is an LCSIS if and only if there is no other longer ISM detected between them.

**Definition 5** All Common Sub-IS (ACSTIS). ACSIS counts all the ISM of two interest sequences, including empty ISM.

LCSIS and ACSIS both provide shared common information between two users’ interest sequences. Intuitively, two users are more similar to each other if they have a longer LCSIS and more ACSIS. The similarity calculation based on LCSIS and ACSIS will be given in the following section. Therefore, the recommendation task addressed in this paper can be defined as the IS-based rating prediction problem, which consists of estimating the utility of items for users using the ratings of their nearest neighbors, who are ranked according to their similarity based on users’ LCSTIS and ACSTIS.

### The New Recommendation Method

**Similarity Based on LCSIS and ACSIS**

In this paper, similarities between users are calculated considering users’ interest sequences. As defined in the previous section, IS can be regarded as time series data that consists of pairs of items and their ratings sorted by time sequence as indicated by the index. As an important indication of the similarity relationship between time series data, LCS has become one of the most commonly used indications [32][33][41]. Recently, Hui Wang [32] has verified that ACS also contains some common information to a certain degree, and ACS-based similarity is competitive for classification problems. Therefore, we assume that users who have longer LCSIS and more ACSIS should also have more similarity in their preferences. The effectiveness of similarity based on LCS and ACS in traditional online recommendation systems should be studied. However, the existing LCS and ACS algorithms cannot satisfy the need to discover the length of LCSIS and the count of ACSIS directly, as they do not incorporate the items and the rating values between two items at the same time in the matching process. Therefore, two extended methods are proposed based on the algorithms proposed in [32][38].

Consider the two IS $\text{isu}_u$ and $\text{isu}_v$ extracted from the rating history of users $u$ and $v$, respectively, where $|\text{isu}_u| = m$ and $|\text{isu}_v| = n$. Let $\omega$ be an $(m + 1) \times (n + 1)$ matrix. Then, the length of LCSIS between $u$’s IS and $v$’s IS, denoted by $|\text{lcsis}(u, v)|$, can be calculated in Eq (2):

$$
|\text{lcsis}(u, v)| = \omega[i, j] = \begin{cases} 
0, & \text{if } i = 0 \text{ or } j = 0 \\
\omega[i - 1, j - 1] + 1, & \text{if } i^u = i^v = \text{frdev}(r^u_i, r^v_i) \leq \theta \\
\max(\omega[i - 1, j], \omega[i, j - 1]), & \text{if } i^u \neq i^v \text{ or frdev}(r^u_i, r^v_i) > \theta 
\end{cases}
$$

where $0 \leq i \leq m$ and $0 \leq j \leq n$. Then, $|\text{lcsis}(u, v)| = \omega[m, n].$

**Example 1.** Consider two interest sequences $\text{isu}_u$ and $\text{isu}_v$ in Table 1 with $\text{Items} = \{C, D, E, I\}$. For any item $i \in \text{Items}$, let $\max(r^u_i) = \max(r^v_i) = 5.0$ and $\min(r^u_i) = \min(r^v_i) = 0.0$. Besides, let $\theta = 0.2$. The set of LCSIS of $\text{isu}_u$ and $\text{isu}_v$ is $\{C \rightarrow I, E \rightarrow I\}$ and thus $|\text{lcsis}| = 2$. The calculation

Collaborative Filtering Recommendation on Users’ Interest Sequences

PLOS ONE | DOI:10.1371/journal.pone.0155739 May 19, 2016 5/17
Collaborative Filtering Recommendation on Users’ Interest Sequences

Table 1. Example of two interest sequences.

|   | $t_1$   | $t_2$   | $t_3$   | $t_4$   |
|---|---------|---------|---------|---------|
| $i_s$ | (C,2.5) | (E,3.0) | (D,4.5) | (I,0.5) |
| $i_r$ | (E,4.0) | (D,2.5) | (C,3.5) | (I,1.5) |

doi:10.1371/journal.pone.0155739.t001

of $|lcsis(u, v)|$ can be performed as below:

$$|lcsis(u, v)| = \omega[4, 4]$$

$$= \omega[3, 3] + 1$$

$$= \max(\omega[3, 2], \omega[2, 3]) + 1$$

$$= \max(\max(\omega[3, 1], \omega[2, 2]), \omega[2, 3]) + 1$$

$$= \max(\max(\max(\omega[3, 0], \omega[2, 1]), \omega[2, 2]), \omega[2, 3]) + 1$$

$$= \max(\max(0, \omega[1, 0] + 1), \omega[2, 2]), \omega[2, 3]) + 1$$

$$= \max(\max(1, \omega[2, 2]), \omega[2, 3]) + 1$$

$$= \max(\max(1, \max(\omega[2, 1], \omega[1, 2])), \omega[2, 3]) + 1$$

$$= \max(\max(1, \max(1, \max(\omega[1, 1], \omega[0, 2])), \omega[2, 3]) + 1$$

$$= \max(\max(1, \max(1, \max(\omega[1, 0], \omega[0, 1])), \omega[2, 3]) + 1$$

$$= \max(1, \max(\omega[2, 2], \omega[1, 3])) + 1$$

$$= \max(1, \max(1, \omega[0, 2] + 1)) + 1$$

$$= 2$$

The count of ACSIS between $i_s$ and $i_r$, denoted by $|acsis(u, v)|$ (this method is not suitable for interest sequences containing repetitive items, and most recommendation systems do not consider repetitive items in the datasets either) can be calculated in the following equation Eq (3):

$$|acsis(u, v)| = \omega[i, j] = \begin{cases} 
1, & \text{if } i = 0 \text{ or } j = 0 \\
\omega[i - 1, j], & \text{if } i, j > 0, \xi_s(j, (it^u_i, r^u_i)) = 0 \\
\omega[i - 1, j] + \omega[i - 1, \xi_s(j, (it^u_i, r^u_i)) - 1], & \text{if } i, j > 0, \xi_s(j, (it^u_i, r^u_i)) > 0
\end{cases}$$

where $0 \leq i \leq m$ and $0 \leq j \leq n$. Then, $|acsis(u, v)| = \omega[m, n]$.

Note that, different from the traditional LCS and ACS algorithms, the calculations of $|lcsis(u, v)|$ and $|acsis(u, v)|$ have to take the deviation between two users’ ratings on the matching items into consideration. The function $\xi_s(j, (it^u_i, r^u_i))$ shown as part of formula (3) obtains the position $x(1 \leq x \leq j)$ where there exists an IP $(it^u_i, r^u_i)$ in $i_r$. Thus, that $it^u_i = it^u_j$ and
The function is given by Eq (4):

$$\xi(x, (it^u, r^u)) = \begin{cases} x, & \text{if } it^u = it^v, f_{adv}(r^u, r^v) \leq \theta \\ 0, & \text{otherwise} \end{cases}$$

where $1 \leq x \leq j$.

Example 2. Consider two interest sequences $is_u$ and $is_v$ in Example 1. The set of ACSIS of $is_u$ and $is_v$ is $\{\phi, C, E, I, C \rightarrow I, E \rightarrow I\}$ and then $|acsis(u, v)| = 6$. The calculation of $|acsis(u, v)|$ can be performed as below:

\[
|acsis(u, v)| = \omega[4, 4] \\
= \omega[3, 4] + \omega[3, 4 - 1] \\
= \omega[2, 4] + \omega[3, 3] \\
= \omega[1, 4] + \omega[1, 1 - 1] + \omega[3, 3] \\
= \omega[0, 4] + \omega[0, 3 - 1] + 1 + \omega[3, 3] \\
= 1 + 1 + 1 + \omega[2, 3] \\
= 3 + \omega[1, 3] + \omega[1, 1 - 1] \\
= 3 + \omega[0, 3] + \omega[0, 3 - 1] + 1 \\
= 4 + 1 + 1 \\
= 6
\]

To compare the similarities between two users’ IS, the normalization of $|lcsis(u, v)|$ and $|acsis(u, v)|$ is conducted in Eqs (5) and (6) as follows:

\[
sim_{lcsis}(u, v) = \frac{|lcsis(u, v)|}{\sqrt{|lcsis(u, u)| \times |lcsis(v, v)|}}
\]

\[
sim_{acsis}(u, v) = \frac{|acsis(u, v)|}{\sqrt{|acsis(u, u)| \times |acsis(v, v)|}}
\]

Then, we can use the factor $\alpha$ to combine these two types of similarity measurements as defined in Eq (7):

\[
sim_{\alpha}(u, v) = \alpha \times \exp^{\alpha_{lcsis}(u, v)} + (1 - \alpha) \times \exp^{\alpha_{acsis}(u, v)}
\]

where the factor $\alpha \in [0, 1]$. The factors $\alpha$ and $(1 - \alpha)$ represent the weights of LCSIS and ACSIS in the similarity measurement, respectively. The value of $\alpha$ should be set to optimize the recommendation results in different applications.

An Updated Similarity by Combining IS-based Similarity

We combine our IS-based similarity measure with traditional similarity measures used in existing collaborative filtering recommendation algorithms. We intend to use the user characteristics contained in IS to further improve the recommendation performance. For the Pearson similarity in Eq (8) (for user-based recommender systems, the Pearson similarity outperforms
other measures of users’ proximity [42]), the similarity measurement to integrate IS-based similarity is defined in Eq (9):

$$pc(u, v) = \frac{\sum_{i \in I_{pu}} (r_u^i - \bar{r})(r_v^i - \bar{r})}{\sqrt{\sum_{i \in I_{pu}} (r_u^i - \bar{r})^2} \sqrt{\sum_{i \in I_{pv}} (r_v^i - \bar{r})^2}}$$ (8)

$$sim(u, v) = pc(u, v) \times f(sim_{IS}(u, v))$$ (9)

where \(pc(u, v)\) is the method to calculate Pearson similarity, and \(f(sim_{IS})\) is a weighting function used to reflect the influence of users’ IS on users’ similarity, as given in Eq (10):

$$f(sim_{IS}) = \frac{\text{comm}(u, v) \times sim_{IS}(u, v)}{\text{total}(u, v)}$$ (10)

where \(\text{comm}(u, v)\) represents the number of common items that user \(u\) and user \(v\) have both rated and \(\text{total}(u, v)\) represents the total of all items that \(u\) and \(v\) have rated. The similarity based on IS is \(sim_{IS}(u, v)\), as defined in the previous section.

The Prediction of Users’ Rating Based on IS

After calculating users’ similarities, we rank all the other users that have rated the target item according to their similarities with the target user and then select the top \(K\) users as the target user’s neighbors for the target item. For the target user \(u\), let \(I_{pred}^u\) be the items to which the recommender system needs to give predicted ratings for the target user, and let \(N_{ui}\) be \(u\)’s neighbors on the target item \(i \in I_{pred}^u\). The equation to predict the rating is shown below in Eq (11):

$$\bar{r}_i^u = \bar{r} + \frac{\sum_{v \in N_{ui}} sim(u, v)(r_v^i - \bar{r})}{\sum_{v \in N_{ui}} sim(u, v)}$$ (11)

where \(\bar{r}_i^u\) is the predicted rating given by the user \(u\) to the item \(i\), \(v\) is one of the nearest neighbors, \(r_v^i\) is \(v\)’s past rating for item \(i\), and \(\bar{r}\) and \(\bar{r}_i\), respectively, represent the average of \(u\)’s and \(v\)’s past ratings.

Experimental Results and Analysis

To verify the effectiveness of our proposed method, we conduct comprehensive experiments on the real datasets and compare the results with some recommendation methods regarding their performance on two common evaluation metrics. In addition, the influence of the rating deviation constraint threshold \(\theta\) and the weights \(\alpha\) of LCSIS and ACSIS in similarity calculations with the top \(K\) neighbors on the results is discussed.

Dataset Description

For this paper, we need to choose appropriate datasets which contain rating timestamps. Moreover, the datasets must have been collected over a long enough period to contain sufficient interest sequences. For these reasons, this paper uses the following datasets: MovieLens 100k and latest small, Flixster, and Ciao.

The MovieLens dataset [16] was collected by GroupLens Research from the MovieLens website. The MovieLens data sets were collected over various periods of time, depending on their size. For our experiments, we used the dataset MovieLens 100K, which contains 100,000 anonymous ratings of approximately 943 movies made by 1682 MovieLens users with rating timestamps spanning from 19 April, 1997 to 22 April, 1998 and MovieLens latest small, which
contains 100,023 ratings applied to 8,552 movies by 706 users between April 03, 1996 and January 09, 2016.

The Flixster with timestamps dataset [17] contains 100,000 ratings to 11,011 movies by 2,048 users during the period from November 2005 to November 2009.

The Ciao with timestamps dataset [18] contains 36,065 ratings to 16,861 products by 2,248 users during the period from July 2000 to November 2013.

**Evaluation Metrics**

To make the experimental results comparable and reproducible, we adopt two well-known metrics, RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error), to measure the accuracy of the predictions. These two measures (\( rmse \) and \( mae \)) can be calculated as shown in Eqs (12) and (13):

\[
rmse = \sqrt{\frac{\sum_{u \in Users} \sum_{i \in I_u^{pred}} (r_{iu} - p_{iu})^2}{\sum_{u \in Users} |I_u^{pred}|}}
\]

\[
mae = \frac{\sum_{u \in Users} \sum_{i \in I_u^{pred}} |r_{iu} - p_{iu}|}{\sum_{u \in Users} |I_u^{pred}|}
\]

where \( u \in Users \) denotes the user, \( i \in I_u^{pred} \) denotes one of \( u \)'s unrated items, \( r_{iu} \) denotes the real rating given by \( u \) to \( i \), \( p_{iu} \) denotes the predicted rating given by \( u \) to \( i \), and \( |I_u^{pred}| \) denotes the number of \( u \)'s unrated items. It is clear that lower \( rmse \) and \( mae \) values indicate a better recommendation accuracy.

**Experimental Setup**

To evaluate the performance of our method and the effectiveness of IS for recommendations, we compare it with the traditional user-based CF recommendation algorithm and a recommendation algorithm based on users’ dynamic information from [8]:

1. User-based CF (UCF) [43]: is a comparative algorithm that uses the rating history of users to calculate the similarities between them and then makes automatic predictions based on those similarities and neighbors’ ratings.

2. The recommendation based on users’ dynamic information (UDI) [8]: is a comparative algorithm that takes users’ dynamic interests into consideration by introducing a decreasing time function to model users’ dynamic interest features.

In our experiments, we refer to our proposed method as ISCF. The existing dataset is split into a training set (80%) and a test set (20%). To avoid overfitting problems, we conduct 5-fold cross-validation experiments. Our work is implemented based on LibRec [44], which is a GPL-licensed Java library for recommender systems.

**Experimental Comparison for Three Methods on Four Datasets**

This section compares the results of the experiments using three methods on four datasets. To comprehensively compare the results, the \( K \) number of neighbors was set to 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. To compare our results with UDI’s best results, some experiments for optimizing UDI’s results were also conducted. As described in [8], the time decay rate \( \lambda \) should be optimized. From the results, we find that UDI performs best in terms of MAE and RMSE.
when $\lambda$ is 0.1 for four datasets. In addition, for our method ISCF, the rating deviation constraint threshold $\theta$ was set to 0.8 and the weight $\alpha$ of LCSIS in similarity calculation was set to 0.5. Figs 1 and 2 show the results of three methods’ MAE and RMSE on the dataset Ciao. From Figs 1 and 2, it can be seen that three methods perform nearly steadily with increasing numbers of neighbors. ISCF is the best and UDI performs even worse than UCF.

The results of three methods’ MAE and RMSE on the dataset Flixster are shown in Figs 3 and 4. It shows that the precision of rating prediction for three methods is the worst when $K = 10$, and then is improved significantly when 10 more neighbors are introduced. Afterwards, the precision increases slightly and then keeps nearly the same with more neighbors’ ratings utilized. Besides, ISCF has the best performance in the three methods no matter how many neighbors are involved in predicting users’ ratings.

The comparison of three methods’ MAE and RMSE on the dataset MovieLens 100k is shown in Figs 5 and 6. It demonstrates that ISCF and UDI perform better than UCF, and in Fig 5, UDI’s MAE approaches ISCF’s when the number of neighbors is more than 80. However, UDI still performs not as well as ISCF in RMSE. Based on the results in Figs 5 and 6, it can be verified that ISCF outperforms UCF and UDI on the dataset MovieLens 100k.

Figs 7 and 8 give three methods’ results on the dataset MovieLens latest small. It can be seen that UDI performs the worst except the results when $K = 10$ and ISCF has the most precise output.

To summarize, our proposed method ISCF has the best performance in predicting users’ ratings to unknown items in all the four datasets. Although UDI performs better than UCF on the dataset MovieLens 100k, it performs the worst on the other three datasets. Therefore, the effectiveness of interest sequences for improving precision of the recommendation results is verified.
Fig 2. Comparison of three methods’ RMSE on Ciao.

doi:10.1371/journal.pone.0155739.g002

Fig 3. Comparison of three methods’ MAE on Flixster.

doi:10.1371/journal.pone.0155739.g003
Fig 4. Comparison of three methods’ RMSE on Flixster.
doi:10.1371/journal.pone.0155739.g004

Fig 5. Comparison of three methods’ MAE on MovieLens 100k.
doi:10.1371/journal.pone.0155739.g005
Fig 6. Comparison of three methods’ RMSE on MovieLens 100k.
doi:10.1371/journal.pone.0155739.g006

Fig 7. Comparison of three methods’ MAE on MovieLens latest small.
doi:10.1371/journal.pone.0155739.g007
The Influence of $\theta$ and $\alpha$

In order to evaluate the influence of LCSIS and ACSIS on the recommendation results, we adopt different values of the rating deviation constraint threshold $\theta$ and the weight $\alpha$ of LCSIS for our proposed method. $\theta$ was set to 0.2, 0.5 and 0.8, which represent weak, medium and strong constraint for users’ rating deviations, respectively. Meanwhile, $\alpha$ was set to 0.2, 0.5 and 0.8 to denote the weight of LCSIS in calculating users’ similarities based on interest sequences. The expected results were supposed to be that different values of $\theta$ and $\alpha$ lead to different recommendation precision. However, the recommendation precision nearly keeps the same. After investigating users’ similarities for different $\theta$ and $\alpha$, it is found that users’ similarities only have tiny fluctuations (e.g. in the dataset MovieLens latest small, the similarities for different $\theta$ and $\alpha$ between the user 599 and the user 194 in Table 2). The reason is that users’ rating history is of high sparsity and thus the length of LCSIS and the count of ACSIS cannot have significant changes for different $\theta$ and $\alpha$. Nevertheless, the length of LCSIS and the count of ACSIS are effective in improving the recommendation precision as shown in experimental comparisons for three methods on four datasets.

Table 2. Example of two users’ similarities in MovieLens latest small for different $\theta$ and $\alpha$.

|       | $\sigma = 0.2$ | $\sigma = 0.5$ | $\sigma = 0.8$ |
|-------|----------------|----------------|----------------|
| $\theta = 0.2$ | 0.2270 | 0.2271 | 0.2273 |
| $\theta = 0.5$ | 0.2270 | 0.2271 | 0.2273 |
| $\theta = 0.8$ | 0.2271 | 0.2272 | 0.2273 |

Fig 8. Comparison of three methods’ RMSE on MovieLens latest small.

doi:10.1371/journal.pone.0155739.g008
Conclusion and Future Work

This paper proposed a new recommendation method based on users’ interest sequences to capitalize on the evolutionary patterns and time sequences of users’ preferences to improve recommendations. First, we formally defined the problem and provided a definition for interest sequences, LCSIS and ACSIS. Then, a similarity measurement method considering LCSIS and ACSIS was proposed. Furthermore, LCSIS and ACSIS-based similarity was combined with traditional similarity to obtain users’ similarity based on users’ interest sequences. In addition, given the IS-based similarity, we select the top $K$ most similar neighbors to predict users’ ratings on unknown items. Finally, comprehensive experiments using four datasets validated the effectiveness of IS in improving recommendation accuracy. Moreover, the experimental results also demonstrated that our method outperforms the traditional user-based CF method and time-aware recommendation method.

In future research work, we will continue to study users’ dynamic behavior patterns and analyze characteristics of their IS to improve the accuracy of recommendation systems by taking other contextual information into account. In addition, we plan to develop new methods that consider users’ social relationships and items’ inner connections. Moreover, the calculation efficiency for determining the LCSIS and ACSIS values remains a challenge.

Supporting Information

S1 File. The detailed data of experiment results.
(XLS)

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants No. 61272186 and No. 61472095.

Author Contributions

Conceived and designed the experiments: WC GY. Performed the experiments: WZ. Analyzed the data: HD. Contributed reagents/materials/analysis tools: YD. Wrote the paper: WC.

References

1. Zi-Ke Z, Chuang L, Yi-Cheng Z, Tao Z. Solving the cold-start problem in recommender systems with social tags. EPL (Europhysics Letters). 2010; 92(2).
2. Tian Q, Zi-Ke Z, Guang C. Information Filtering via a Scaling-Based Function. PLoS ONE. 2013; 8(5).
3. Yu L, Liu C, Zhang Z. Multi-linear interactive matrix factorization. Knowledge Based Systems. 2015: 307–315.
4. Koren Y. Collaborative Filtering with Temporal Dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD’09. New York, NY, USA: ACM; 2009. p. 447–456. Available from: http://doi.acm.org/10.1145/1557019.1557072.
5. Campos PG, Diez F, Cantador I. Time-aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols. User Modeling and User-Adapted Interaction. 2014 Feb; 24 (1–2):67–119. Available from: http://dx.doi.org/10.1007/s11257-012-9136-x, doi: 10.1007/s11257-012-9136-x
6. Ding Y, Li X. Time Weight Collaborative Filtering. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. CIKM’05. New York, NY, USA: ACM; 2005. p. 485–492. Available from: http://doi.acm.org/10.1145/1099554.1099689.
7. Ding Y, Li X, Orlowska ME. Recency-based Collaborative Filtering. In: Proceedings of the 17th Australasian Database Conference—Volume 49. ADC’06. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.; 2006. p. 99–107. Available from: http://dl.acm.org/citation.cfm?id=1151736.1151747.
8. Cheng J, Liu Y, Zhang H, Wu X, Chen F. A New Recommendation Algorithm Based on User’s Dynamic Information in Complex Social Network. Mathematical Problems in Engineering. 2015; 2015:6. doi: 10.1155/2015/281629

9. Wang Y, Yin G, Cai Z, Dong Y, Dong H. A trust-based probabilistic recommendation model for social networks. Journal of Network and Computer Applications. 2015; 55(5):59–67. Available from: http://www.sciencedirect.com/science/article/pii/S108480451500079X. doi: 10.1016/j.jnca.2015.04.007

10. Ying JJC, Lu EHC, Lee WC, Weng TC, Tseng VS. Mining User Similarity from Semantic Trajectories. In: Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks. LBSN ’10. New York, NY, USA: ACM; 2010. p. 19–26. Available from: http://doi.acm.org/10.1145/1867699.1867703.

11. Parenti C, Spaccapietra S, Renso C, Andrienko G, Andrienko N, Bogorny V, et al. Semantic Trajectories Modeling and Analysis. ACM Comput Surv. 2013 Aug; 45(4):42:1–42:32. Available from: http://doi.acm.org/10.1145/2501654.2501656. doi: 10.1145/2501654.2501656

12. Zhong H, Zhang S, Wang Y, Weng S, Shus Y. Mining Users’ Similarity from Moving Trajectories for Mobile E-commerce Recommendation. International Journal of Hybrid Information Technology. 2014; 7(4):309. doi: 10.14257/ijhit.2014.7.4.26

13. Liu H, Schneider M. Similarity Measurement of Moving Object Trajectories. In: Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming. IWGS ’12. New York, NY, USA: ACM; 2012. p. 19–22. Available from: http://doi.acm.org/10.1145/2442968.2442971.

14. Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma WY. Mining User Similarity Based on Location History. In: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS’08. New York, NY, USA: ACM; 2008. p. 34:1–34:10. Available from: http://doi.acm.org/10.1145/1463434.1463477.

15. Zheng Y, Zhang L, Ma Z, Xie X, Ma WY. Recommending Friends and Locations Based on Individual Location History. ACM Trans Web. 2011 Feb; 5(1):5:1–5:44. Available from: http://doi.acm.org/10.1145/1921591.1921596. doi: 10.1145/1921591.1921596

16. Harper F. Maxwell and Konstan Joseph A. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TIIIS). 2015 Dec; 5(4). Available from http://dx.doi.org/10.1145/2827872.

17. Jamali Mohsen and Ester Martin. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks In: Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys’10); 2010.

18. Tang, Jiliang and Gao, Huiji and Liu, Huan. mTrust: discerning multi-faceted trust in a connected world. In: Proceedings of the fifth ACM international conference on Web search and data mining; 2012.

19. Zhongqi L, Zhicheng D, Jianxun L, Xing X, Qiang Y. Content-based collaborative filtering for news topic recommendation. In: Procedings of the twenty-Ninth AAAI Conference on Artificial Intelligence. Austin Texas, USA; 2015. p. 217–223.

20. Goldberg D, Nichols D, Oki BM, Terry D. Using Collaborative Filtering to Weave an Information Tapestry. Commun ACM. 1992 Dec; 35(12):61–70. Available from: http://doi.acm.org/10.1145/138859.138867. doi: 10.1145/138859.138867

21. Koren Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD’08. New York, NY, USA: ACM; 2008. p. 426–434. Available from: http://doi.acm.org/10.1145/1401880.1401944.

22. Jamali S. Probabilistic Models for Recommendation in Social Networks [(Thesis) Ph.D.]. Applied Sciences: School of Computing Science, Simon Fraser University; 2013.

23. kumar Bokde D, Girase S, Mukhopadhyay D. Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey. International Journal of Advance Foundation and Research in Computer (IJAFRC). 2014; 1(6):6. Available from: http://arxiv.org/abs/1503.07475.

24. Salakhutdinov R, Mnih A. Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. In: Proceedings of the 25th International Conference on Machine Learning. ICML’08. New York, NY, USA: ACM; 2008. p. 860–867. Available from: http://doi.acm.org/10.1145/1390156.1390267.

25. Ekstrand MD, Riedl JT, Konstan JA. Collaborative Filtering Recommender Systems. Foundations and Trends in Human-Computer Interaction. 2010; 4(2):81–173. Available from: http://dx.doi.org/10.1561/1100000009. doi: 10.1561/1100000009

26. Shi Y, Larson M, Hanjalic A. Collaborative Filtering Beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. ACM Comput Surv. 2014 May; 47(1):3:1–3:45. Available from: http://doi.acm.org/10.1145/2556270. doi: 10.1145/2556270
27. Adomavicius G, Tuzhilin A. Context-aware Recommender Systems. In: Proceedings of the 2008 ACM Conference on Recommender Systems. RecSys’08. New York, NY, USA: ACM; 2008. p. 335–336. Available from: http://doi.acm.org/10.1145/1454008.1454068.

28. Nakatsuji M, Fujiwara Y, Uchiyama T, Toda H. Collaborative Filtering by Analyzing Dynamic User Interests Modeled by Taxonomy. In: Proceedings of the 11th International Conference on The Semantic Web—Volume Part I. ISWC’12. Berlin, Heidelberg: Springer-Verlag; 2012. p. 361–377. Available from: http://dx.doi.org/10.1007/978-3-642-35176-1_23.

29. Lee TQ, Park Y, Park YT. A Time-based Approach to Effective Recommender Systems Using Implicit Feedback. Expert Syst Appl. 2008 May; 34(4):3055–3062. Available from: http://dx.doi.org/10.1016/j.eswa.2007.06.031. doi: 10.1016/j.eswa.2007.06.031

30. Yin H, Cui B, Chen L, Hu Z, Huang Z. A Temporal Context-aware Model for User Behavior Modeling in Social Media Systems. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. SIGMOD’14. New York, NY, USA: ACM; 2014. p. 1543–1554. Available from: http://doi.acm.org/10.1145/2588555.2593685.

31. Chen J, Wang C, Wang J. A Personalized Interest-Forgetting Markov Model for Recommendations. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence; 2015. p. 16–22. Available from: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9387.

32. Wang H. All Common Subsequences. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. IJCAI’07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2007. p. 635–640. Available from: http://dl.acm.org/citation.cfm?id=1625275.1625377.

33. Deorowicz S, Obstoj J. Constrained Longest Common Subsequence Computing Algorithms in Practice. Computing and Informatics / Computers and Artificial Intelligence. 2010; 29:427–445.

34. Mishra R, Kumar P, Bhasker B. A web recommendation system considering sequential information. Decision Support Systems. 2015; 75(0):1–10. Available from: http://www.sciencedirect.com/science/article/pii/S0167923615000755. doi: 10.1016/j.dss.2015.04.004

35. Tsai YT. The constrained longest common subsequence problem. Information Processing Letters. 2003; 88(4):173–176. Available from: http://www.sciencedirect.com/science/article/pii/S002001900300406X. doi: 10.1016/j.ipl.2003.07.001

36. Fang L, Wan M, Yu M, Yan J, Liu Z, Wang P. Analysis of similarity measure in the longitudinal study using improved longest common subsequence method for lung cancer. Biomedical Signal Processing and Control. 2015; 15(0):60–66. Available from: http://www.sciencedirect.com/science/article/pii/S1746809414001451. doi: 10.1016/j.bspc.2014.09.010

37. Wang H. Subsequence Counting As a Measure of Similarity for Sequences. International Journal of Pattern Recognition and Artificial Intelligence. 2007; 21(04):745–758. Available from: http://www.worldscientific.com/doi/abs/10.1142/S021800140700565X. doi: 10.1142/S021800140700565X

38. Lin Z, Wang H, McClean S. A Multidimensional Sequence Approach to Measuring Tree Similarity. IEEE Transactions on Knowledge & Data Engineering. 2012 February; 24(2):197–208. doi: 10.1109/TKDE.2010.239

39. Greenberg RI. Fast and Simple Computation of All Longest Common Subsequences. CoRR. 2002;cs.DS/0211001. Available from: http://arxiv.org/abs/cs.DS/0211001.

40. Egho E, Raissi C, Calders T, Jay N, Napoli A. On Measuring Similarity for Sequences of Itemssets. Data Min Knowl Discov. 2015 May; 29(3):732–764. Available from: http://dx.doi.org/10.1007/s10618-014-0362-1. doi: 10.1007/s10618-014-0362-1

41. Kurbalija V, Radovanovic M, Geler Z, Ivanović M. The Influence of Global Constraints on Similarity Measures for Time-series Databases. Know-Based Syst. 2014 Jan; 56:49–67. Available from: http://dx.doi.org/10.1016/j.knosys.2013.10.021. doi: 10.1016/j.knosys.2013.10.021

42. Herlocker JL, Konstan JA, Borchers A, Riedl J. An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR’99. New York, NY, USA: ACM; 1999. p. 230–237. Available from: http://doi.acm.org/10.1145/312624.312682.

43. Su X, Khoshgoftaar TM. A Survey of Collaborative Filtering Techniques. Adv in Artif Intell. 2009 Jan; 2009:4:2–4:2. Available from: http://dx.doi.org/10.1155/2009/421425. doi: 10.1155/2009/421425

44. Guo G, Zhang J, Sun Z, Yorke-Smith N. LibRec: A Java Library for Recommender Systems. In: Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP2015); 2015.