

Research Article

Investigating the Temporal Effect of User Preferences with Application in Movie Recommendation

Wen-Jun Li,^{1,2} Qiang Dong,¹ and Yan Fu¹

¹Big Data Research Center & Web Sciences Center, University of Electronic Science and Technology of China, Chengdu 611731, China

²Suzhou Institute of Industrial Technology, Suzhou 215104, China

Correspondence should be addressed to Qiang Dong; dongq1984@qq.com

Received 3 January 2017; Accepted 21 February 2017; Published 12 March 2017

Academic Editor: Salvatore Carta

Copyright © 2017 Wen-Jun Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the rapid development of mobile Internet and smart devices, more and more online content providers begin to collect the preferences of their customers through various apps on mobile devices. These preferences could be largely reflected by the ratings on the online items with explicit scores. Both of positive and negative ratings are helpful for recommender systems to provide relevant items to a target user. Based on the empirical analysis of three real-world movie-rating data sets, we observe that users' rating criterions change over time, and past positive and negative ratings have different influences on users' future preferences. Given this, we propose a recommendation model on a session-based temporal graph, considering the difference of long- and short-term preferences, and the different temporal effect of positive and negative ratings. The extensive experiment results validate the significant accuracy improvement of our proposed model compared with the state-of-the-art methods.

1. Introduction

Nowadays, a huge ecosystem of independent content providers (such as Facebook, Netflix, Google Maps, and Snapchat) and consumers (web users) is emerging on the mobile Internet. Confronted with the problem of finding a needle in the haystack, many web users usually resort to information filtering technology to find more relevant contents. Nowadays, recommender systems have been deployed on the websites of many industries [1], to make the web services more suitable and engaged to their users and promote the scale and profitability of such businesses [2]. In the recent decades, recommender systems have received considerable research attention in the literature, and many effective recommendation approaches have been proposed, such as social network-based recommendation models [3], graph-based recommendation models [4, 5], and context aware recommendation models [6, 7]; a recent and up-to-date review can be found in the works of Lu et al. [8].

Many of these works are focused on movie recommendation or based on movie-rating data sets [9, 10]. Typically, in online video-watching websites with recommender systems,

users are asked to rate movies with discrete scores to express their individual opinions, where a high score usually indicates user preference on this movie. Take <https://www.netflix.com> as an example, users are suggested to rate movies and TV shows (items in general) in a rating scale from 1 star to 5 stars, where one star means "Hate It," and five stars mean "Love It." This kind of explicit feedbacks can largely reflect user preferences. Even if a user dislikes a movie after watching it, he might be attracted by its title, cast, director, genres, or others; otherwise he would never watch it. Hence, the negative ratings indicate many useful information and thus should not be neglected or simply considered to be negative. Many works have shown that both of positive and negative opinions are effective to make effective recommendations.

First, given a rating scale where the highest score denotes the most positive opinion and the lowest score indicates the most negative opinion, users' rating scores do not distribute evenly along the whole rating scale [11]. Second, different users may have different rating criterions, some good-tempered users are willing to give high scores whereas other critical people seldom give full marks to any items they have watched [12]. Last but not least, the negative ratings indicate

TABLE 1: Basic statistics of three data sets. The sparsity is defined as $|E|/(m * n)$, where $|E|$ is the number of ratings and m and n denote the numbers of users and items.

Data set	# users	# items	# links	Sparsity	Rating scale
MovieLens	943	1682	100000	6.3×10^{-2}	1-5
Netflix	3000	3000	97367	1.08×10^{-2}	1-5
MovieTweatings	2583	911	67040	2.85×10^{-3}	1-10

dislike and simultaneously relevance, and they may play an either negative or positive role depending on the sparsity of training set and the popularity of the corresponding items [13].

As the mobile platforms become more and more user friendly, computationally powerful, and readily available, online content providers have begun to develop mobile apps to offer more personalized contents. People can watch their favorite movies and TV shows wherever and whenever they have a break. This mobile feature poses a new challenge to recommendation systems. Most of previous works do not consider the temporal difference in the rating criterions of users. According to the memory effect of movie watching behavior [14] and the anchoring bias phenomena of movie-rating behavior [15], the current rating of one user will be influenced by his previous watching and rating history. Therefore, individual rating criterions may vary in different periods, depending on the previous items he has watched. Besides, a user's negative ratings may have a different temporal influence from his positive ratings on his future preference.

In this paper, we empirically analyze three typical data sets created by popular online video services (MovieLens, Netflix, and MovieTweatings) with focus on the temporal effects of the rating behavior of each individual user. We concentrate on the time-varying rating criterion and different temporal effects of positive and negative ratings on future behavior. We propose a session-based recommendation model taking into account these temporal characteristics of user ratings. Compared with five state-of-the-art methods on the aforementioned movie-rating data sets, our proposed model is validated to give more accurate prediction of user preference.

2. Empirical Analysis

In this section, we analyze empirically the temporal difference in users' rating criterions and the temporal effects of positive and negative ratings, with the hope of understanding the temporal characteristics of users' rating behaviors and verifying the following two assertions.

Assertion I. The rating criterion of a user varies over time.

Assertion II. The positive and negative ratings of a user have different temporal influences on his future preference.

For the convenience of readers, we list all the notations used in this paper in "Notations."

2.1. Data Sets. Three real-world data sets, MovieLens, Netflix, and MovieTweatings [18], are employed in this paper. The

MovieLens data were collected by the GroupLens Research Project at University of Minnesota, through the MovieLens website (<https://movielens.umn.edu>) during the seven-month period from September 19, 1997, to April 22, 1998. It consists of 943 users and 1682 movies, and 10^5 discrete rating records from 1 star to 5 stars. The Netflix data is random sampling from the original Netflix data set provided by <http://Netflixprize.com>. It is composed of 97367 rating scores of 3000 users on 3000 movies from January 2005 to May 2005. The MovieTweatings is a data set consisting of ratings on movies contained in well-structured tweets on Twitter. By removing users who rate less than 10 movies from the snapshots with 100 K records, the MovieTweatings data set consists of 67040 ratings of 2583 users on 911 movies from February 28, 2013, to September 2, 2013. The basic statistics of these data sets are presented in Table 1.

2.2. The Rating Criterion. In this paper, we investigate users' rating criterions in two aspects: *average rating score* and *rating scale*. Specifically, we consider the monthly average rating score and rating scale of each user as two independent random variables and estimate their standard deviations across months. To obtain a reliable estimation, we consider only the users who are active in more than 2 months of the whole period. Figures 1(a), 1(b), and 1(c) show the distributions of standard deviation of average rating scores for MovieLens, Netflix, and MovieTweatings data sets, respectively. The mean values of the deviations for three data sets (0.36427, 0.56429, and 0.79849) are all significantly greater than 0 (p value ~ 0 , obtained by one-sided t -test). Similarly, Figures 1(d), 1(e), and 1(f) show the distributions of standard deviation of users' rating scales. The mean values for MovieLens, Netflix, and MovieTweatings are 0.93737, 0.93495, and 1.05155, respectively (p value ~ 0 , obtained by one-sided t -test). These observations indicate that every user has a significantly changing rating criterion over time, *empirical evidence of Assertion I.*

2.3. The Positive and Negative Ratings. Note that the rating criterion on items varies from person to person; we take the median score $\bar{r}_i = (r_i^{\max} + r_i^{\min})/2$ of each individual user u_i , instead of the median score of systematic rating scale, to distinguish his own *positive rating* (rating score no less than \bar{r}_i) and *negative rating* (rating score less than \bar{r}_i).

We use *session* to represent a continuous period of user activity; thus the records of user u_i can be divided into several sequential sessions $S_i = \{S_{i,t_1}, S_{i,t_2}, \dots, S_{i,t_n}\}$. In this paper, the sessions are divided by month; that is to say, two ratings of the same user are in the same session if and only if they occur in the same month. For a user u_i , the items rated with positive ratings by user u_i in session S_{i,t_k} constitute his *positive item*

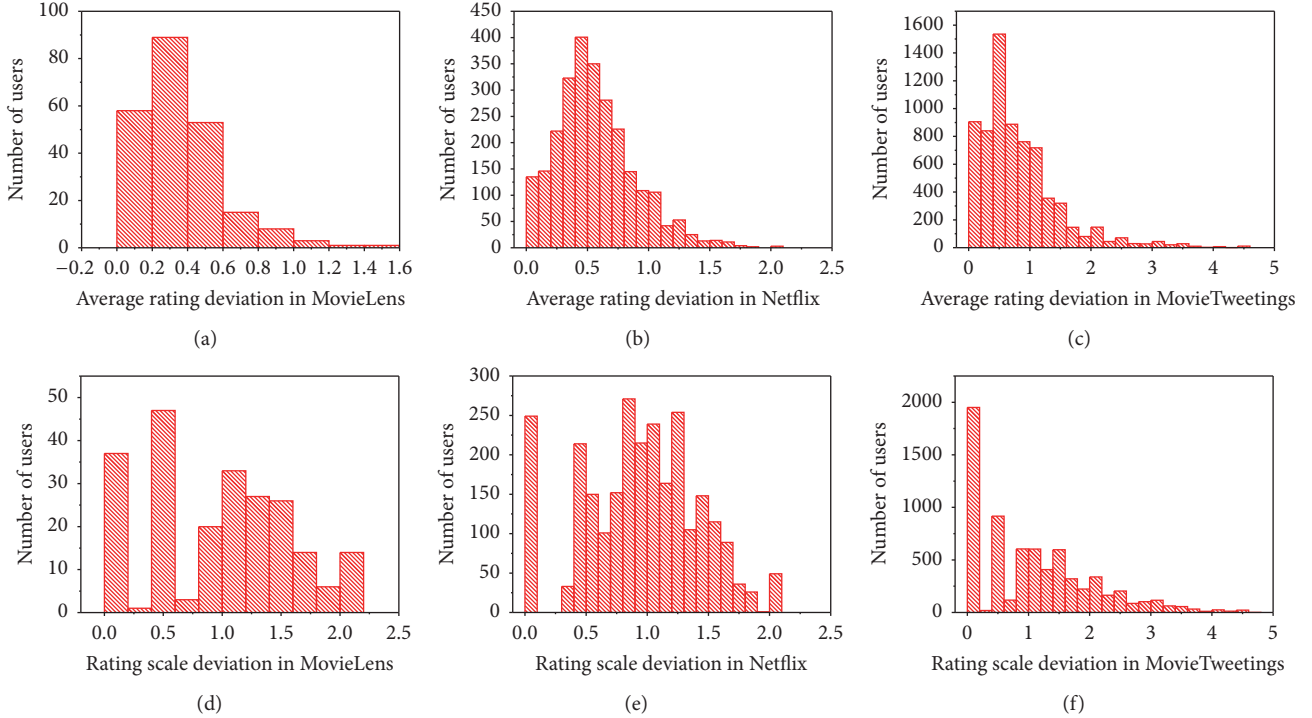


FIGURE 1: The distributions of standard deviation of average rating scores and users' rating scales for MovieLens, Netflix, and MovieTweatings data sets, respectively.

set P_{i,t_k} , and the *negative item set* N_{i,t_k} is similarly defined. For a target user u_i , we take his latest positive item set P_{i,t_n} as future preference, and all the previous positive and negative item sets P_{i,t_k} and N_{i,t_k} are treated as previous interests, $k = 1, 2, \dots, n-1$.

The correlation $\text{Cor}(A, B)$ between two item sets A and B is defined as the averaged cosine similarity on all item pairs, where one item is from A and the other is from B . Figure 2 plots the correlation $\text{Cor}(P_{i,t_n}, P_{i,t_k})$ (the black line) and $\text{Cor}(P_{i,t_n}, N_{i,t_k})$ (the red line) against the time gap $|t_n - t_k|$, of course averaged over all users, for MovieLens, Netflix, and MovieTweatings data sets. We can see that the future preference of a user is clearly more influenced by his positive ratings than the negative ratings in the past. From the temporal point of view, the bigger the time gap is, the less influenced the future preference is by the previous positive/negative ratings. However, the decay rates of influences of positive and negative opinions vary for different data sets. For MovieLens, the influence of positive ratings on future preference is more stable than negative ratings ($\sigma_{\text{positive}} = 0.00584$, $\sigma_{\text{negative}} = 0.02325$), while, for Netflix, the decay rates of influences of positive and negative ratings are very similar to each other ($\sigma_{\text{positive}} = 0.00747$, $\sigma_{\text{negative}} = 0.00912$).

Since the first and the last sessions contain data of only 1 day and 2 days for MovieTweatings data set, we ignore the last points of curves with time gap of 6 months. Different from above observations, we find that the influence of negative ratings is more stable than that of positive ratings in MovieTweatings ($\sigma_{\text{positive}} = 0.01174$, $\sigma_{\text{negative}} = 0.00319$). Therefore, users' positive and negative ratings have different

temporal influences on his future preferences, *empirical evidence of Assertion II*.

3. Recommendation Model

Based on the session-based temporal graph (STG) introduced by Xiang et al. [4], we propose a session-based recommendation model with the temporal effect of user preferences (STeUP), which is an enhanced version of the Injected Preference Fusion (IPF) model associated with STG. Users and items are represented by user nodes $u \in U$ and movie nodes $o \in O$, respectively. To represent users' ratings at different periods, we associate a session node S_{i,t_k} to the movies rated by user u_i in this session. These three types of nodes are connected by weighted directed edges, namely, $E_{U,O}$, $E_{O,U}$, $E_{O,S}$, and $E_{S,O}$. The edges affiliated to session nodes reflect the short-term rating criterions of users, while the edges affiliated to user nodes reflect users' long-term preferences. Figure 3 gives an example of session-based temporal graph.

To eliminate the effect of different rating criterions of different individuals, the rating score of a user is normalized according to his own rating scale:

$$R_{i\alpha} = \frac{r_{i\alpha} - r_i^{\min}}{r_i^{\max} - r_i^{\min}}, \quad (1)$$

which reflects users' long-term rating criterions. In this way, the rating scores of all users can be strictly regulated to $[0, 1]$, where the maximum rating score of each user is set to 1 and the minimum rating score is fixed on 0.

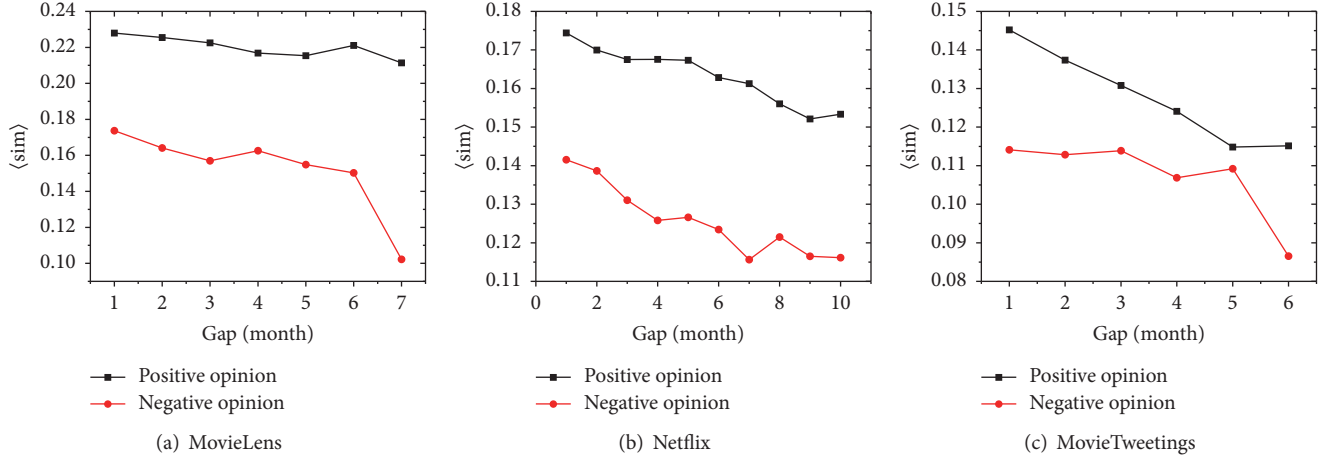


FIGURE 2: The temporal influence of user's positive and negative opinions for MovieLens, Netflix, and MovieTweatings. The black one and red one represent user's positive and negative opinions, respectively.

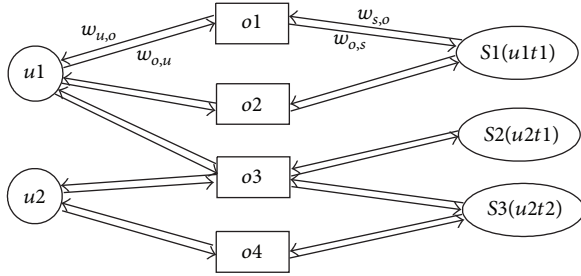


FIGURE 3: The example of session-based temporal graph with rating information.

Since the short-term rating criterion of a user varies at different periods, we normalize his rating score in a particular session S_{i,t_k} by

$$R_{i\alpha}^{S_{i,t_k}} = \frac{r_{i\alpha} - r_{S_{i,t_k}}^{\min}}{r_{S_{i,t_k}}^{\max} - r_{S_{i,t_k}}^{\min}}. \quad (2)$$

Recall that our recommendation task is to recommend movies for a target user to watch in the future. Of course, the rating whose occurrence time is closer to the target time T is more useful to the recommendation task. Since the temporal influences of positive and negative ratings may be different, following previous works [19, 20], we use two exponential functions to model the relevance of positive and negative ratings at time $t_{i\alpha}$ with user's preference at the target time T . Hence, edge weights in $E_{U,O}$ and $E_{O,U}$ are defined as

$$w_{i,\alpha} = w_{\alpha,i} = \begin{cases} R_{i\alpha} \cdot e^{-\lambda(T-t_{i\alpha})} & \text{if } R_{i\alpha} \geq \frac{r_i^{\max} + r_i^{\min}}{2} \\ R_{i\alpha} \cdot e^{-\mu(T-t_{i\alpha})} & \text{if } R_{i\alpha} < \frac{r_i^{\max} + r_i^{\min}}{2}. \end{cases} \quad (3)$$

Similarly in a specified session, the rating whose occurrence time is closer to the target time T is more important in this session. We use the same exponential functions to model

the temporal influences of positive and negative ratings in a session, and the median rating value in this session is taken to distinguish positive and negative ratings. Thus, the edge weights of $E_{S,O}$ and $E_{O,S}$ are calculated by

$$w_{S_{i,t_k},\alpha} = w_{\alpha,S_{i,t_k}} = \begin{cases} R_{i\alpha}^{S_{i,t_k}} \cdot e^{-\lambda(T-t_k)} & \text{if } R_{i\alpha}^{S_{i,t_k}} \geq \frac{r_{S_{i,t_k}}^{\max} + r_{S_{i,t_k}}^{\min}}{2} \\ R_{i\alpha}^{S_{i,t_k}} \cdot e^{-\mu(T-t_k)} & \text{if } R_{i\alpha}^{S_{i,t_k}} < \frac{r_{S_{i,t_k}}^{\max} + r_{S_{i,t_k}}^{\min}}{2}. \end{cases} \quad (4)$$

After setting the initial edge weights of STG, we normalize these edge weights as follows:

$$\psi(v, v') = \begin{cases} \frac{w_{v,v'}}{\sum_{x \in N_v} w_{v,x}} & \text{if } v \in U \cup S, v' \in O \\ \frac{\eta w_{v,v'}}{\eta \sum_{x \in N_v \cup U} w_{v,x} + \sum_{k \in N_v \cap S} w_{v,x}} & \text{if } v \in O, v' \in U \\ \frac{w_{v,v'}}{\eta \sum_{x \in N_v \cup U} w_{v,x} + \sum_{k \in N_v \cap S} w_{v,x}} & \text{if } v \in O, v' \in S. \end{cases} \quad (5)$$

A larger η indicates that users' long-term preferences play a more important role in preference propagation.

Given a target user u_i , the basic idea of the preference propagation is to first inject initial preference on both the user node u_i and his latest session node and then propagate the preference to candidate movie nodes through various paths in the graph. As defined in [4], the preference propagated by each path p is the production of the initial preference $\gamma(v_0)$ assigned to the target user node u_i (or the latest session node $S_{i,n}$) and the weights of all edges on the path:

$$\phi(p) = \prod_{e_{v,v'} \in p} w_{v,v'} \gamma(v_0). \quad (6)$$

$\gamma(v_0)$ depends on the node type:

$$\gamma(v_0) = \begin{cases} \beta & \text{if } v_0 = u_i \\ 1 - \beta & \text{if } v_0 = s_{i,n}, \end{cases} \quad (7)$$

where $\beta = 0$ means no preference is injected into the user node, while $\beta = 1$ means no preference is injected into the session node. Similar to the previous work [4], we consider only the shortest paths (distance = 3) from source node to unknown movie nodes, which can be obtained effectively by Bread-First-Search. Consequently, we use $P_{i\alpha}$ to represent the set of shortest paths from source nodes to an unknown movie node o_α for user u_i , and the estimated preference $\phi_{i\alpha}$ of user u_i on movie o_α is then measured as

$$\phi_{i\alpha} = \sum_{p \in P_{i\alpha}} \phi(p), \quad (8)$$

where $\phi(p)$ is the weight of path p defined as (6). The top-ranked movies sorted by preference value are then recommended.

4. Experiment Results

4.1. Evaluation Metrics. In order to predict users' future preferences based on the past interaction records, all the records are listed in ascending order of rating time. We take the records occurred in the latest 30 days as the probe set E^P and the remaining records as the training set E^T for all data sets. The training set is treated as known information, while no information from the probe set is allowed to be used for recommendation. Moreover, we denote the latest time among the training set as the target time T . In this paper, four typical metrics are employed to evaluate the accuracy, diversity, novelty, and coverage of recommendation results.

4.1.1. Accuracy. Accuracy is one of the most important evaluation metrics of a recommendation system. Both Precision and Recall could be used to measure the accuracy of the recommendation. Precision P is the fraction of recommended items that are relevant, while Recall R is defined as the ratio of the number of relevant items in the recommendation list to the number of preferred items in the probe set. However, Precision and Recall seem to be two sides of the seesaw; that is to say, given a fixed length of recommendation list, when one end rises, the other end falls. The $F1$ measure is proposed to find a suitable trade-off between Precision and Recall, which is defined as

$$F1 = \frac{2 \times P \times R}{P + R}, \quad (9)$$

where $P = (1/|U|) \sum_{i=1}^{|U|} (h_i/L)$ and $R = (1/|U|) \sum_{i=1}^{|U|} (h_i/n_{ip})$, where $|U|$ is the number of users, h_i is the number of relevant items in the recommendation list of u_i , and n_{ip} is the number of all preferred items in the probe set of user u_i . Generally speaking, for a given length L of recommendation list, the method with higher $F1$ value is the better one.

4.1.2. Diversity. Diversity is used to measure the difference between recommendation lists of different users. An excellent algorithm should recommend as widely distributed items as possible, because people are glad to get personalized suggestions. We use the Hamming distance H_{ij} to measure the diversity of recommendation lists,

$$H_{ij} = 1 - \frac{q_{ij}}{L}, \quad (10)$$

where q_{ij} is the number of common items in the recommendation lists of user u_i and user u_j . $H_{ij} = 0$, if u_i and u_j get identical recommendation list consisting of L items. Diversity is defined as the mean value of Hamming distance:

$$\text{Diversity}(L) = \frac{1}{|U|(|U| - 1)} \sum_{i \neq j} H_{ij}. \quad (11)$$

4.1.3. Novelty. Novelty quantifies the capacity of a method to generate novel and unexpected recommendations, which may be greatly contributed by less popular items (i.e., items of low degree) that are unlikely to be known previously. It can be simply measured as the average degree of the recommended items. Specifically, for a target user u_i whose top- L recommendation list is denoted by L_i , his novelty is defined as [21]

$$\text{Novelty}_i(L) = \frac{1}{L} \sum_{\alpha \in L_i} k_\alpha. \quad (12)$$

Averaging over the novelty of all users, we obtain the novelty of the system.

4.1.4. Coverage. Coverage measures the percentage of items that an algorithm is able to recommend to users. It can be calculated as the ratio of the number of distinct items in the users' lists to the total number of items in the system, which reads

$$\text{Coverage} = \frac{1}{|O|} \sum_{i=\alpha}^n \delta_\alpha, \quad (13)$$

where $|O|$ is the number of items in set O , $\delta_\alpha = 1$ only if item o_α is recommended to at least one user (i.e., o_α is in at least one user's list), and otherwise $\delta_\alpha = 0$. Undoubtedly, recommending more popular items will result in lower coverage.

4.2. Parameter Adjustment. Before comparing the proposed model with the baseline methods, we investigate the impacts of the parameters λ , μ , β , and η on the performance of the STeuP model. As we see in Section 2.3, the temporal effect of positive and negative opinions may be different in different online websites. Thus, we first examine the effect of parameters λ and μ , which govern the decay rates of temporal influence of positive and negative opinions on users' future preference. The bigger the parameters λ and μ are, the less affected the future behaviors are by users' past positive and negative opinions. Without loss of generality, we set $\eta = 1$ and $\beta = 0.5$ when tuning λ and μ .

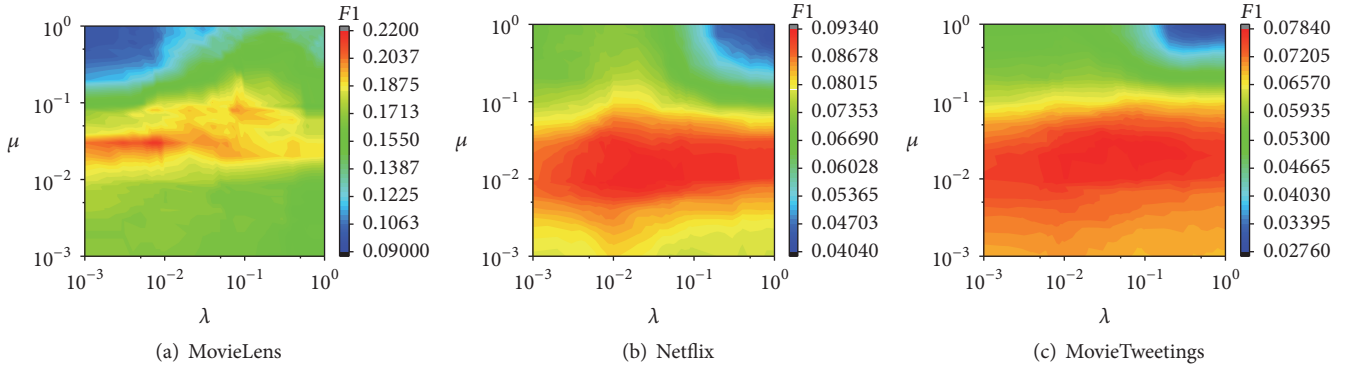


FIGURE 4: The heat map of $F1$ against parameters λ and μ on MovieLens, Netflix, and MovieTweatings.

TABLE 2: Performance comparison of STeuP model distinguishing the temporal influence of positive and negative ratings on MovieLens, Netflix, and MovieTweatings.

Data set	λ	μ	$F1$	HD	Novelty	Coverage
MovieLens	0.0	0.0	0.1697	0.7946	306.37	0.1058
	0.007	0.03	0.2148	0.7647	275.18	0.1285
Netflix	0.0	0.0	0.0770	0.6848	600.66	0.2976
	0.009	0.009	0.0930	0.6757	555.41	0.3067
MovieTweatings	0.0	0.0	0.0689	0.6869	381.43	0.2084
	0.09	0.02	0.0783	0.6582	344.52	0.2100

Figures 4(a), 4(b), and 4(c) plot the heat map of $F1$ against parameters λ and μ for MovieLens, Netflix, and MovieTweatings, respectively. The $\log-\lambda$ is along the x -axis while the y -axis is for $\log-\mu$. The different $F1$ values along the z -axis are indicated by different colors. Firstly, we observe that the $F1$ is more sensitive to μ than λ ; that is, given a fixed value of λ , the changing range of $F1$ is much bigger when traversing the parameter μ . Secondly, the results on all data sets show that there is an obvious “ridge” along the x -axis, where we can get the optimal $F1$ value. Hence, we can firstly fix λ to a small value ($10^{-2} \sim 10^{-3}$) and tune the parameter μ to find the local optimal value, and then fix μ and adjust λ to find the globally optimal accuracy.

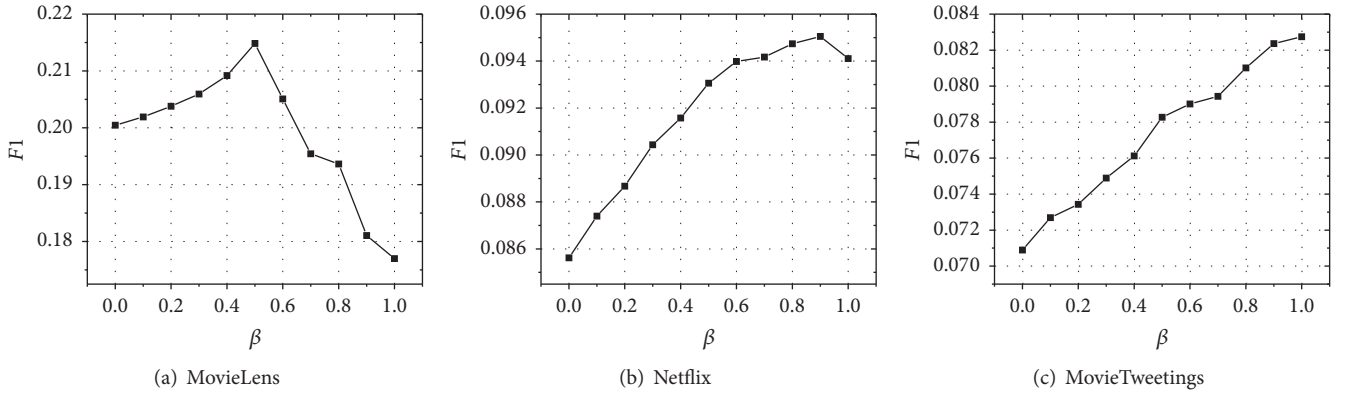
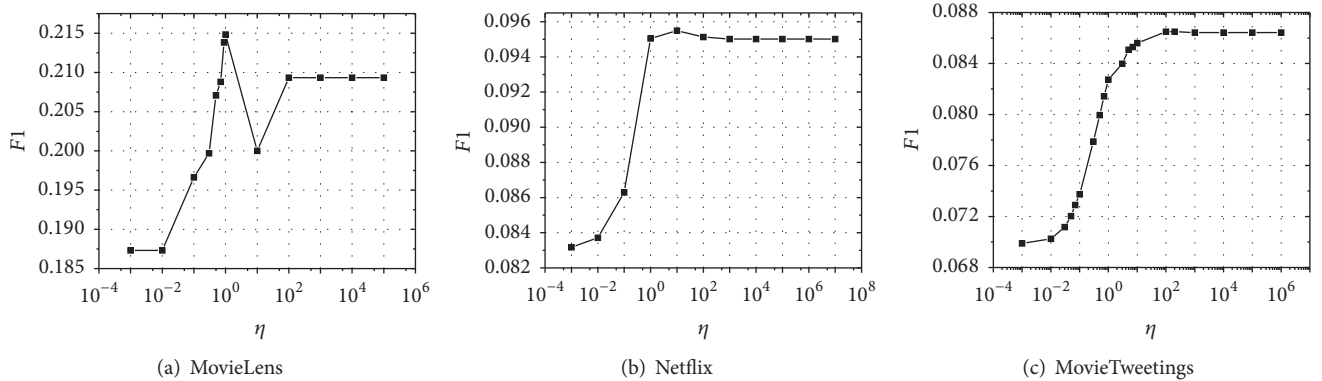
By setting the values of λ and μ to 0, we can get the recommendation results without temporal influence, which are presented in Table 2. We can see that consideration of temporal influence by weighting users’ positive and negative opinions with different temporal decay rates leads to performance improvements. From the values of parameters λ and μ when we get the optimal $F1$, we can find that the decay rate of temporal effect of positive opinions is much smaller than that of negative opinions on MovieLens, but it is bigger than that of negative opinions for MovieTweatings. For Netflix, the decay rates of positive and negative opinions are almost the same. This validates again our inference on the different temporal influences of positive and negative opinions on three data sets in Section 2.3.

In our STeuP model, parameter β controls the ratio of injected preferences into user nodes against session nodes. If β equals 0, no preference is injected into the user node; if β equals 1, no preference is injected into the session node. Thus, parameter β is used to balance the effect of long-term and

short-term interests in the initial phase, where the larger β is, the stronger the influence of long-term preferences is. The results of how accuracy changes against β for three data sets are shown in Figure 5. Firstly, the results show that ignoring long-term preferences ($\beta = 0$) cannot generate good results. Secondly, in a sparser data set, the β value for the optimal $F1$ is bigger. Generally speaking, optimal results can be obtained by combing long-term and short-term interests together. In the next discussion, we fix β to 0.5, 0.9, and 1.0 for MovieLens, Netflix, and MovieTweatings data sets, respectively.

Parameter η is used to balance the influence of long-term and short-term preferences in the process of preference propagation. $\eta = 0$ means item nodes are only connected to users nodes and item-item similarity depends only on users’ long-term preference and vice versa. Figures 6(a), 6(b), and 6(c) plot the change of $F1$ against parameter η on three data sets. As the x -axis is for the logarithmic value of η , we can see that, for MovieLens and Netflix data sets, a parameter η close to 1 is corresponding to the optimal $F1$, while the optimal value of η for MovieTweatings is close to 10^2 . This observation verifies that both users’ long-term and short-term opinions are important to measure item similarity. Furthermore, users’ long-term opinions are more important than short-term opinions on sparse data sets.

4.3. Comparison of Methods. We compare our proposed model with other five models listed in Table 3. The mark \checkmark in the table indicates whether a model distinguishes user ratings as positive and negative opinions and/or considers temporal influence. *UCF* is an important collaborative filtering method which calculates the similarity between users based on the rating information. *NBI* is a network-based

FIGURE 5: The $F1$ against parameter β on MovieLens, Netflix, and MovieTweatings.FIGURE 6: $F1$ against parameter η on MovieLens, Netflix, and MovieTweatings.TABLE 3: The six recommendation models for comparison in this paper. The mark \checkmark under PN or T indicates whether the model distinguishes opinions as positive and negative opinions and/or considers temporal influence.

Method	Description	PN	T
UCF	User-based CF [16]	—	—
UOS	User-based CF combining the user opinion spreading process [12]	\checkmark	—
NBI	Network-based inference algorithm on binary data [17]	—	—
SNBI	NBI with different weights for positive and negative opinions [13]	\checkmark	—
IPF	Personalized random walk on STG [4]	—	\checkmark
S T euP	The proposed model in Section 3	\checkmark	\checkmark

inference algorithm which can provide more efficient and effective recommendation than collaborative filtering. Both of the methods do not distinguish user ratings as positive and negative opinions or consider temporal influence. *UOS* is an improved user-based collaborative filtering model which estimates the similarity between users based on the spreading process of users' positive and negative opinions on user-item bipartite networks. *SNBI* is the enhanced version of *NBI* which assigns two different weights to the positive and negative opinions. *UOS* and *SNBI* distinguish users' ratings as positive and negative opinions, but do not utilize temporal influence. *IPF* is the recommendation model proposed together with session-based temporal graph, which is based on binary data and distinguishes users' long-term and short-term preferences. *S T euP* is our proposed model based on the two assertions in Section 2, which takes into account both

of the temporal differences of users' rating criterions and the different temporal effects of users' positive and negative ratings.

What is more, we also check the performance of a famous matrix factorization recommendation algorithm [22], which is very successful in rating prediction with the help of time information. However, the $F1$ value of this method on the aforementioned three data sets is all less than 0.001, perhaps because it does not apply to the situation of binary preference prediction. Hence, we do not present the results of the matrix factorization methods for comparison in this paper.

Given the recommendation length $L = 10$, the recommendation performance of these six methods for MovieLens, Netflix, and MovieTweatings is reported in Tables 4, 5, and 6, respectively.

TABLE 4: Comparison of different recommendation models on MovieLens data set. For UOS, $\lambda = 0.2$; for SNBI, $\lambda = -2.1$; for MS-IPF, $\beta = 0.1$, $\rho = 1.5$, and $\eta = 1.0$; for STeuP, $\beta = 0.5$, $\lambda = 0.007$, $\mu = 0.03$, and $\eta = 1.0$. The length of recommendation list is $L = 10$.

Algorithm	F1	HD	Novelty	Coverage
UCF	0.0761	0.9399	224.63	0.2854
UOS	0.0839	0.8735	268.10	0.1691
NBI	0.1081	0.7724	319.62	0.0873
SNBI	0.1144	0.8382	287.57	0.1353
IPF	0.1649	0.8010	281.11	0.1667
STeuP	0.2147	0.7647	275.18	0.1285

TABLE 5: Comparison of different recommendation models on Netflix data set. For UOS, $\lambda = 0.0$; for SNBI, $\lambda = -1.3$; for MS-IPF, $\beta = 0.5$, $\rho = 1.1$, and $\eta = 10$; for STeuP, $\beta = 0.9$, $\lambda = 0.009$, $\mu = 0.009$, and $\eta = 1.0$. The length of recommendation list is $L = 10$.

Algorithm	F1	HD	Novelty	Coverage
UCF	0.0732	0.8876	453.55	0.2720
UOS	0.0733	0.7638	546.90	0.1666
NBI	0.07524	0.5908	677.00	0.0942
SNBI	0.0794	0.6057	657.37	0.1441
IPF	0.0783	0.6607	636.08	0.2084
STeuP	0.095	0.6169	589.88	0.2358

TABLE 6: Comparison of different recommendation models on MovieTweatings data set. For UOS, $\lambda = -0.2$; for SNBI, $\lambda = -0.9$; for MS-IPF, $\beta = 0.1$, $\rho = 1.5$, and $\eta = 1.0$; for STeuP, $\beta = 1.0$, $\lambda = 0.09$, $\mu = 0.02$, and $\eta = 100$. The length of recommendation list is $L = 10$.

Algorithm	F1	HD	Novelty	Coverage
UCF	0.0633	0.7966	335.17	0.1185
UOS	0.0490	0.7785	337.87	0.1837
NBI	0.0706	0.4881	468.30	0.0440
SNBI	0.0725	0.4999	463.43	0.0542
IPF	0.0785	0.5024	462.35	0.0447
STeuP	0.0865	0.5269	394.51	0.1341

- (i) Collaborative filtering methods (UCF and UOS) can provide higher diversity, novelty, and coverage, but lower accuracy than graph-based methods (NBI and SNBI).
- (ii) With the help of distinguishing users' ratings into positive and negative opinions, both memory-based CF method and graph-based method can provide more accurate recommendation. The $F1$ values of UOS and SNBI are both better than UCF and NBI, especially for the graph-based method. The diversity, novelty, and coverage of SNBI are all improved compared with NBI. However, for sparse data sets (MovieTweatings), UOS and SNBI cannot work well as expected. The results of SNBI are almost the same with that of NBI, while $F1$ of UOS is even worse than UCF.
- (iii) The IPF model, which considers users' long-term and short-term temporal influence, can provide quite good recommendation results for all data sets, except for the accuracy for Netflix.
- (iv) The proposed method STeuP model performs the best with regard to accuracy for all data sets. Besides, it can provide better diversity, novelty, and coverage than

graph-based methods, especially for the sparse data sets (Netflix and MovieTweatings).

5. Conclusions

In this paper, we analyzed users' dynamic rating criteria and we discovered the different temporal effects of user's positive and negative opinions on future preference. We propose a session-based movie recommendation model using this dynamic rating information, which simultaneously considers the temporal change of users' rating criteria and different temporal effects of positive and negative ratings. The extensive experiments on three real-world movie data sets show that the accuracy of our proposed model is significantly improved compared to the state-of-the-art methods.

Notations

- u, U : A user, user set
- o, O : A movie, movie set
- S_i : The session set of user u_i
- S_{i,t_k} : The k -th session of user u_i
- $r_{i\alpha}$: The rating of user u_i on movie o_α

r_i^{\max}, r_i^{\min} :	The highest and lowest rating scores made by user u_i
\bar{r}_i :	The middle value of user u_i 's ratings
P_{i,t_k} :	The positive item set in S_{i,t_k}
N_{i,t_k} :	The negative item set in S_{i,t_k}
$r_{S_{i,t_k}}^{\max}, r_{S_{i,t_k}}^{\min}$:	The highest and lowest scores in S_{i,t_k}
$E_{X,Y}$:	The edge sets from node set X to node set Y
h_i :	The number of relevant items (namely, the items collected by the user u_i in the probe set) in the recommendation list of u_i
n_{ip} :	The number of selected items in user u_i 's probe set
$t_{i\alpha}$:	The time stamp when user u_i rates movie o_α
λ, μ :	The decay factors controlling the extent of temporal influences of positive and negative ratings
$N_\nu, N_\nu \cap S, N_\nu \cap U$:	The neighbor set, neighbor session set, and neighbor user set of item ν
η :	The parameter used to adjust the preference propagation of an item node to its user neighbors or session neighbors
ν, p :	A node and a path on the session-based graph
$w_{\nu,\nu'}$:	The weight of edge $e_{\nu,\nu'}$ in path p
$\gamma(\nu_0)$:	The value of injected preferences on the source node ν_0
β :	The parameter used to tune the ratio of injected preferences on the user node against the session node.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

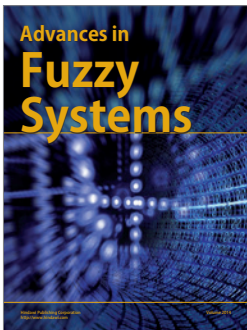
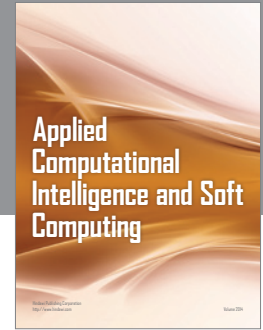
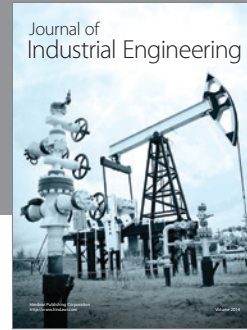
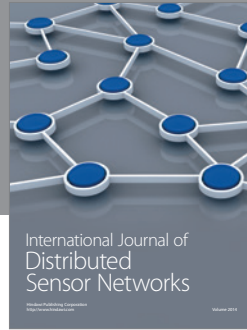
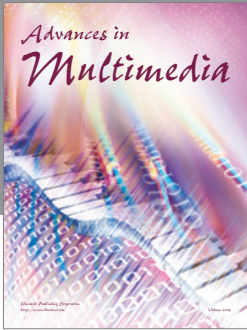
Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no. 61673085) and Science & Technology Department of Sichuan Province (Grant no. 2016GZ0081).

References

- [1] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "A generalized taxonomy of explanations styles for traditional and social recommender systems," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 555–583, 2012.
- [2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [3] M. Tinghua, J.-J. Zhou, M.-L. Tang et al., "Social network and tag sources based augmenting collaborative recommender system," *IEICE Transactions on Information and Systems*, vol. 98, no. 4, pp. 902–910, 2015.
- [4] L. Xiang, Q. Yuan, S. Zhao et al., "Temporal recommendation on graphs via long- and short-term preference fusion," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 723–731, ACM, Washington, DC, USA, July 2010.
- [5] D.-C. Nie, Y.-H. An, Q. Dong, Y. Fu, and T. Zhou, "Information filtering via balanced diffusion on bipartite networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 421, pp. 44–53, 2015.
- [6] K. Verbert, N. Manouselis, X. Ochoa et al., "Context-aware recommender systems for learning: a survey and future challenges," *IEEE Transactions on Learning Technologies*, vol. 5, no. 4, pp. 318–335, 2012.
- [7] C. Zeng, Q. Wang, S. Mokhtari, and T. Li, "Online context-aware recommendation with time varying multi-armed bandit," in *Proceedings of the the 22nd ACM SIGKDD International Conference (KDD '16)*, pp. 2025–2034, San Francisco, Calif, USA, August 2016.
- [8] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.
- [9] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez, "Social knowledge-based recommender system. Application to the movies domain," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10990–11000, 2012.
- [10] P. Winoto and T. Y. Tang, "The role of user mood in movie recommendations," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6086–6092, 2010.
- [11] Q. Guo, Y. Li, and J.-G. Liu, "Information filtering based on users' negative opinions," *International Journal of Modern Physics C*, vol. 24, no. 5, Article ID 1350032, 2013.
- [12] X.-S. He, M.-Y. Zhou, Z. Zhuo, Z.-Q. Fu, and J.-G. Liu, "Predicting online ratings based on the opinion spreading process," *Physica A: Statistical Mechanics and Its Applications*, vol. 436, pp. 658–664, 2015.
- [13] W. Zeng, Y.-X. Zhu, L. Lü, and T. Zhou, "Negative ratings play a positive role in information filtering," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 23-24, pp. 4486–4493, 2011.
- [14] L. Hou, X. Pan, Q. Guo, and J.-G. Liu, "Memory effect of the online user preference," *Scientific Reports*, vol. 3, Article ID 06560, 2014.
- [15] Z. Yang, Z.-K. Zhang, and T. Zhou, "Anchoring bias in online voting," *Europhysics Letters*, vol. 100, no. 6, Article ID 68002, 2012.
- [16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '94)*, pp. 175–186, ACM, Chapel Hill, NC, USA, October 1994.
- [17] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 4, Article ID 046115, 2007.
- [18] S. Dooms, T. De Pessemier, and L. Martens, "Movietweetings: a movie rating dataset collected from twitter," in *Proceedings of the Workshop on Crowdsourcing and Human Computation for Recommender Systems (CrowdRec-RecSys '13)*, Hong Kong, 2013.
- [19] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05)*, pp. 485–492, November 2005.

- [20] W.-J. Li, Y.-Y. Xu, Q. Dong, J.-L. Zhou, and Y. Fu, "TaDb: a time-aware diffusion-based recommender algorithm," *International Journal of Modern Physics C. Computational Physics and Physical Computation*, vol. 26, no. 9, Article ID 1550102, 2015.
- [21] T. Zhou, L.-L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *EPL*, vol. 81, no. 5, Article ID 58004, 2008.
- [22] L. Xiang and Q. Yang, "Time-dependent models in collaborative filtering based recommender system," *Web Intelligence and Intelligent Agent Technologies*, vol. 1, pp. 450–457, 2009.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

