Journal of Computers Vol. 30 No. 5, 2019, pp. 18-30
doi:10.3966/199115992019103005002

Recommendation with the Cold-Start Problem in Evolving User-Movie Network

Shu-Juan Zhang[1,2], Juan Zhang[3], Zhen Jin[1,3,4*]

[1] School of Mechatronic Engineering, North University of China, Taiyuan 030051, Shanxi, China
mf258368@126.com, jinzhn@263.net

[2] School of Data Science and Technology, North University of China, Taiyuan 030051, Shanxi, China
mf258368@126.com

[3] Complex Systems Research Center, Shanxi University, Taiyuan 030006, Shanxi, China
zhangjuan1020@sxu.edu.cn, jinzhn@263.net

[4] Shanxi Key Laboratory of Mathematical Techniques and Big Data Analysis on Disease Control and Prevention, Shanxi University, Taiyuan 030006, Shanxi, China
jinzhn@263.net

Received 4 October 2017; Revised 14 March 2018; Accepted 3 May 2018

Abstract. Faced with the problem of information overload on the Internet, recommender systems have been paid more and more attention. At present, many recommendation algorithms have been proposed. However, there are two problems in these recommendation methods which must be solved. Most of them are studied in a single round of recommendation, but lots of complex systems in reality are evolving with time. Besides, the cold-start problem has always been intractable in the recommender system. Therefore, in this paper, we propose two recommendation methods (i.e., Method I and Method II) to solve the movie cold-start problem for the evolving user-movie network which is comprised of users and movies, and then examine the recommendation performance and long-term effects of these two methods in evolving networks, and evaluate influence factors through the numerical simulation. The experiment results show that the network structure, the movie attributes and the length of the recommendation list affect the recommendation accuracy and its novelty. The effects of Method I and Method II are better than those of other recommendation methods in previous works. Moreover, in Method Iwe should lay more emphasis on the relation between nodes, while in Method II we should focus on node attributes.

Keywords: cold-start, evolving networks, recommendation, user-movie network

1 Introduction

With the rapid development of the Internet, the explosion of network resources can bring about not only great convenience for our daily life, but also much trouble. For example, people can do shopping or watch movies at home by surfing the Internet, but they will also face plenty of options for products or movies. Therefore, how to get what we really want from a large amount of information on network is the problem today we face. To solve this problem, recommender systems have emerged as information filtering tools, which provide users with the information they may be interested in according to the users' needs and personal hobbies. To date, recommender systems have been applied to many domains [1-3].

At present, many recommendation algorithms have been proposed. Among which collaborative filtering is the most successful [4-6], and it is divided into user-based collaborative filtering and item-based collaborative filtering. Additionally, content-based recommendation has also been widely studied

* Corresponding Author

[7-8]. Combining collaborative filtering and content-based recommendation, some scholars proposed hybrid recommendation [9-10]. With proposition of concept of complex network, the network-based recommendation method was proposed subsequently by some researchers [11-13]. Under this method, recommender systems are represented as a particular class of the networks, which consists of vertices and edges. Generally, it adopts bipartite network where there are two types of vertices: one type represents users, and the other type denotes items. Each edge connecting vertices in the bipartite network shows the interaction between a user and an item [14-15]. The above methods are applied to many specific fields, especially in movie recommendation [16-20]. However, in current movie recommendation research, there are still some problems that need to solve. Firstly, a very important problem in recommender systems is the cold-start problem. For the movie recommender system, new users and new movies inevitably take part in the system. Since they have no any historic record, they cannot be recommended successfully by previous method, and it affects the recommendation quality of the recommender system. In this case, how to provide personalized recommendation for new users, or recommend new movies to users who are likely to be interested in them is a crucial problem that recommendation method should solve. However, only a few works consider the cold-start problem [21-24]. Secondly, most of the existing studies focus only on the performance of the recommendation methods in a single round of recommendation, and it can only evaluate the short-term effects of the methods. In fact, many complex systems are evolving with time. When we study the performance of the recommendation method in the evolving network, we should assess its long-term effect. At present, there is little research to consider the evolution of the network [25-26].

To solve the above two problems, it is also necessary to study the recommendation method with the cold-start problem in evolving networks. In this paper, we take the evolution of user-movie network as an example, and consider the movie cold-start problem. With it, user-movie network structure and movie attributes are simultaneously considered in recommendation methods proposed in this paper. On the one hand, we can recommend movies for each user based on the network structure, which can reflect the users' historical activities. On the other hand, we can make recommendations for each user according to the movie attributes. Focusing on user-based similarity and item-based similarity, respectively, the two recommendation methods are proposed in this paper. The final recommendation score is got by weighting recommendation scores based on the similarity and based on movie attributes to evaluate the long-term effects of the two methods in the evolving network. And, the numerical simulation is adopted to compare two recommendation methods under different indexes.

The rest of this paper is organized as follows: In Section 2, two recommendation methods are proposed. In Section 3, we give two metrics for the performance of recommendation methods. In Section 4, we study the recommendation on the user-movie network by combing real evolution data, and the two methods are compared. Finally, we make a conclusion in Section 5.

2 Algorithms

The main contents of this section are as follows. Firstly, we give the structure of user-movie network. Based on the network structure, the similarity between movies and the similarity between users are calculated. According to two kinds of similarity, we can propose two recommendation methods. For each method, we can calculate a sort of recommendation score based on network structure, denoted by structure-based recommendation score and another recommendation score based on movies attributes, denoted by attribute-based recommendation score. Then, the final recommendation score of this method is got by weighting two kinds of recommendation scores.

In this paper, the user-movie network is a bipartite network that is composed of users and movies. The nodes in the network can be divided into two types: user nodes and movie nodes. The relation between users and movies is illustrated in Fig. 1. The set of users is denoted by $U = \{u_1, u_2, \cdots, u_M\}$, where M is the total number of users. The set of movies is denoted by $O = \{o_1, o_2, \cdots, o_N\}$ and N stands for the total number of movies. For each movie, it has many attributes, but this paper mainly focuses on the three attributes: movie genre, director and actor. The sets of genres, directors and actors are denoted by $C = \{c_1, c_2, \cdots, c_S\}$, $D = \{d_1, d_2, \cdots, d_K\}$ and $A = \{a_1, a_2, \cdots, a_Q\}$, respectively, where S, K and Q denote the total number of genres, directors, chief actors, respectively. The genres of movies contain action, comedy, adventure, thriller, war, etc..

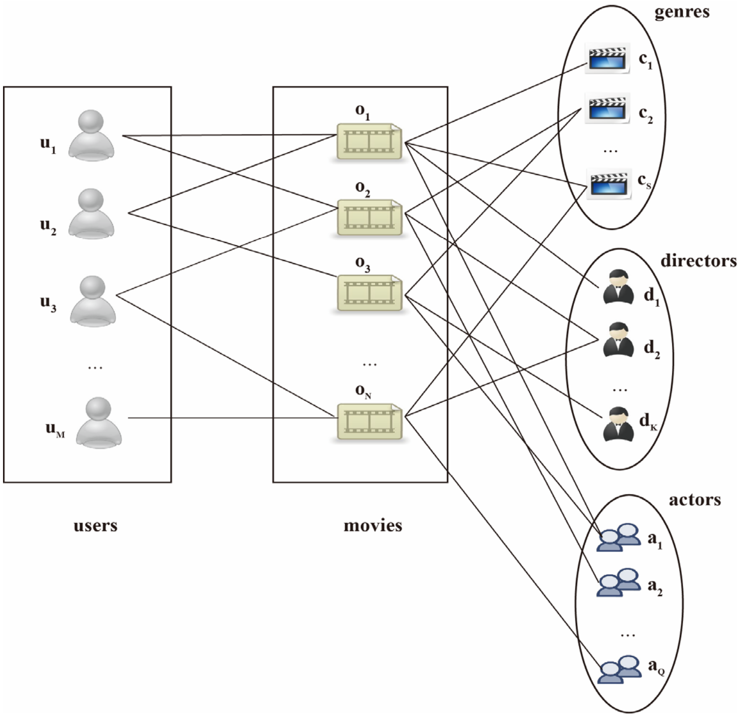


Fig. 1. The schematic diagram of the relation between users, movies and movie attributes. The movie attributes include genre, director and actor

Firstly, based on network structure, we apply the adjacency matrix $R=(r_{i\alpha})_{k_1 \times k_2}$, where $k_1 \le M$ and $k_2 \le N$, to represent the relation between users and movies. If the movie o_α $(\alpha = 1,2,\cdots,N)$ is watched by the user u_i $(i=1,2,\cdots,M)$, $r_{i\alpha}=1$, otherwise $r_{i\alpha}=0$. Then, based on the matrix R, the similarity between the movie o_α and the movie o_β can be calculated as follows.

$$Sim_{item}(o_\alpha,o_\beta)=\frac{\sum_{i=1}^{M} r_{i\alpha}r_{i\beta}}{\sum_{i=1}^{M} r_{i\alpha}+\sum_{i=1}^{M} r_{i\beta}-\sum_{i=1}^{M} r_{i\alpha}r_{i\beta}}. \quad \textbf{(1)}$$

And, the structure-based recommendation score of the unwatched movie o_α for the user u_i can be represented by

$$f_{i\alpha}=\sum_{\beta=1}^{n} Sim_{item}(o_\alpha,o_\beta) r_{i\beta},\ n \le N. \quad \textbf{(2)}$$

Analogously, the similarity between the user u_i and the user u_j can be computed as follows.

$$Sim(u_i,u_j)=\frac{\sum_{\alpha=1}^{n} r_{i\alpha}r_{j\alpha}}{\sum_{\alpha=1}^{n} r_{i\alpha}+\sum_{\alpha=1}^{n} r_{j\alpha}-\sum_{\alpha=1}^{n} r_{i\alpha}r_{j\alpha}} \quad n \le N. \quad \textbf{(3)}$$

And the structure-based recommendation score of the unwatched movie o_α for the user u_i is calculated by

$$f_{i\alpha}=\sum_{j=1}^{M} Sim(u_i,u_j) r_{j\alpha}. \quad \textbf{(4)}$$

Next, we calculate attribute-based recommendation score by investigating the relation between movies and movie attributes (genre, director and actor), which can be represented by adjacency matrix. The relation between movies and genres can be represented by the matrix $G=(g_{\alpha\gamma})_{k_2 \times k_3}$, where $k_3 \le S$. If the movie o_α belongs to the genre c_γ $(\gamma=1,2,\cdots,S)$, $g_{\alpha\gamma}=1$, otherwise $g_{\alpha\gamma}=0$. The matrix $B=(b_{\alpha\eta})_{k_2 \times k_4}$ describes the relation between movies and directors, where $k_4 \le K$. If the movie o_α is directed by the director d_η $(\eta=1,2,\cdots,K)$, $b_{\alpha\eta}=1$, otherwise $b_{\alpha\eta}=0$. The matrix $V=(v_{\alpha q})_{k_2 \times k_5}$ represents the relation between movies and actors, where $k_5 \le Q$. If the actor a_q $(q=1,2,\cdots,Q)$ is the chief actor of the movie o_α, $v_{\alpha q}=1$, otherwise $v_{\alpha q}=0$. Hence, the similarity between the movie o_α and the movie o_β can be represented by

$$Sim(o_\alpha,o_\beta)=Sim_c(o_\alpha,o_\beta)+Sim_d(o_\alpha,o_\beta)+Sim_a(o_\alpha,o_\beta), \tag{5}$$

where

$$Sim_c(o_\alpha,o_\beta)=\frac{\sum_{j=1}^{s} g_{\alpha j}g_{\beta j}}{\sum_{j=1}^{s} g_{\alpha j}+\sum_{j=1}^{s} g_{\beta j}-\sum_{j=1}^{s} g_{\alpha j}g_{\beta j}},\ s \le S, \tag{6}$$

$$Sim_d(o_\alpha,o_\beta)=\frac{\sum_{j=1}^{k} b_{\alpha j}b_{\beta j}}{\sum_{j=1}^{k} b_{\alpha j}+\sum_{j=1}^{k} b_{\beta j}-\sum_{j=1}^{k} b_{\alpha j}b_{\beta j}},\ k \le K, \tag{7}$$

$$Sim_a(o_\alpha,o_\beta)=\frac{\sum_{j=1}^{q} v_{\alpha j}v_{\beta j}}{\sum_{j=1}^{q} v_{\alpha j}+\sum_{j=1}^{q} v_{\beta j}-\sum_{j=1}^{q} v_{\alpha j}v_{\beta j}},\ q \le Q, \tag{8}$$

represent three kinds of similarity according to movie genres, directors and actors, respectively.
And we normalize $Sim(o_\alpha,o_\beta)$ and express it as follows.

$$Sim_{attr}(o_\alpha,o_\beta)=\frac{Sim(o_\alpha,o_\beta)-X_{\min}}{X_{\max}-X_{\min}}, \tag{9}$$

where $X_{\max}$ and $X_{\min}$ are the maximum and minimum values of the original data set before normalization, respectively. Therefore, the attribute-based recommendation score of the movie o_α for the user u_i is calculated as follows.

$$h_{i\alpha}=\sum_{\beta=1}^{n} Sim_{attr}(o_\alpha,o_\beta)r_{i\beta},\ n \le N. \tag{10}$$

As far as the movie degree concerned, which is defined as the number of edges connected to this movie, the movies in the system can be divided into two types, namely, new movies and the movies with the non-zero degree. The new movies have not been watched by any user, which means they have no network structure, so the final recommendation score for the new movie o_α to the user u_i is calculated only by using the movie attributes, and it is expressed as

$$s_{i\alpha}=h_{i\alpha}. \tag{11}$$

For the movie o_α with non-zero degree, it can be recommended by integrating the network structure with the movie attributes, and the final recommendation score to the user u_i can be represented by

$$s_{i\alpha} = w_1 \times f_{i\alpha} + (1 - w_1) \times h_{i\alpha} \, , \tag{12}$$

where w_1 represents the weight of structure-based recommendation score. In the formula (12), if $f_{i\alpha}$ and $h_{i\alpha}$ are respectively calculated by the formula (2) and formula (10), which is denoted by Method I in the following sections; if $f_{i\alpha}$ and $h_{i\alpha}$ are computed by the formula (4) and formula (10), which is denoted by Method II.

3 Metrics for Algorithmic Performance

In order to evaluate the performance of the above two methods in evolving networks, the recommendation accuracy and the novelty are used as two metrics in this paper. Recommendation accuracy is one of the basic metrics. For a user u_i, the precision is defined as

$$P_i(L) = \frac{X_i(L)}{L} \, , \tag{13}$$

where $X_i(L)$ represents the number of the movies that are actually watched by the user u_i in the recommended movies, and L denotes the length of the recommendation list. The precision $P(L)$ of the whole system is obtained by the following formula, whose computing procedure is given in Table 1.

Table 1. Computing $P(L)$

The computing procedure of $P(L)$
Data: original data T for each month and the recommendation list W
Result: $P(L)$ for each month
Given w_1 and L
for each $u_i \in U$ do
$X_i(L) \leftarrow \lvert T(u_i) \cap W(u_i) \rvert$
$P_i(L) \leftarrow \frac{X_i(L)}{L}$
$P(L) \leftarrow P(L) + P_i(L)$
end
$P(L) \leftarrow \frac{P(L)}{M}$

$$P(L) = \frac{1}{M} \sum_{i=1}^{M} P_i(L) \, . \tag{14}$$

Besides the accuracy, the recommendation novelty is a very important metric. For a user u_i, the novelty is defined as

$$N_i(L) = \frac{Y_i(L)}{L} \, , \tag{15}$$

where $Y_i(L)$ denotes the number of new movies that are included in the recommendation list of the user u_i. Then, the novelty $N(L)$ of the whole system is mean value of $N_i(L)$ for all users, whose computing procedure is given in Table 2.

Table 2. Computing $N(L)$

The computing procedure of *N(L)*
Data: new movie list *Z* for each month and the recommendation list *W*
Result: *N(L)* for each month
Given w_1 and L
for each $u_i \in U$ do
$Y_i(L) \leftarrow \lvert Z(u_i) \cap W(u_i) \rvert$
$N_i(L) \leftarrow \frac{Y_i(L)}{L}$
$N(L) \leftarrow N(L) + N_i(L)$
end
$N(L) \leftarrow \frac{N(L)}{M}$

4 Application

4.1 Data

In this section, we apply the real data including time information from Netflix data set to examine the performance of the two methods in evolving networks. These data are grades of movies rated by users, and a movie can be ranked from one to five by each user. The higher the grade is, the more the user likes the movie. We process the original data set and select the users who have made ratings on at least 20 movies, and select the movies whose ratings are higher than 2. By selecting the data, the final data set contains 560 users, 1285 movies and 81539 links, and the time range is from December 1999 to December 2004.

4.2 Network Evolution

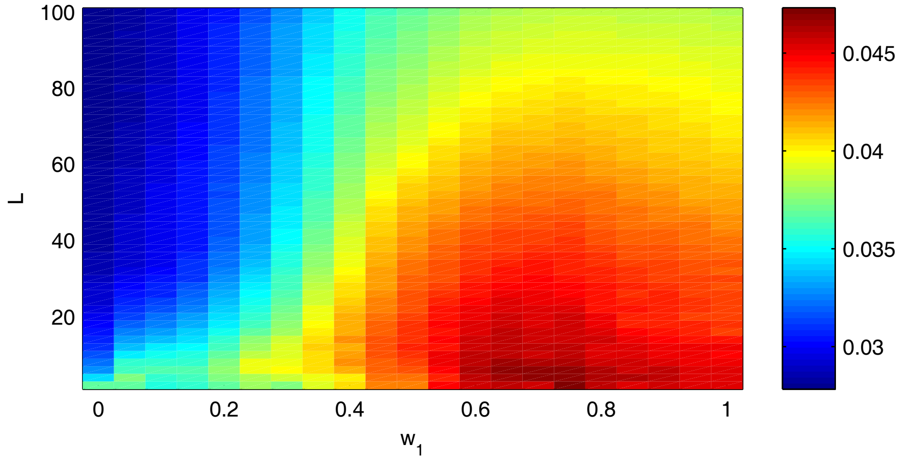
We take the data from December 1999 to April 2004 as training set E^T, and treat the data from May 2004 to December 2004 as the probe set E^P. Firstly, the training set E^T is applied to construct initial user-movie network. Then we adopt Method I and Method II to predict the evolution of the network from May 2004 to December 2004. In the recommendation methods proposed in this paper, it is assumed that L movies are recommended to each user every month, and users accept the recommendation by watching the movies with higher recommendation score in the L movies. In each month, the number of movies to be watched for each user is determined by the probe set. After this, a new network is formed, and the next network evolution will be based on this new one. To evaluate the recommendation performance for each month, we need to compare E^T with E^P in each month.

4.3 Experiments

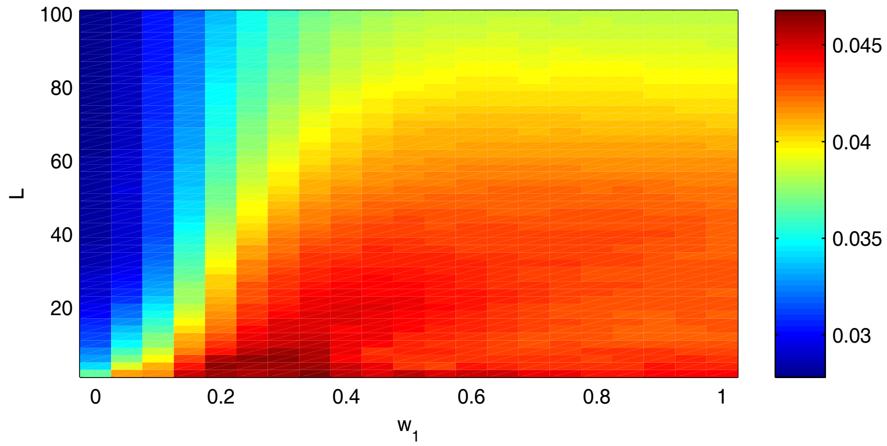
The structure-based recommendation and the attribute-based recommendation are both considered in the two methods, so we can adjust the weight w_1 in the formula (12) to investigate the impacts of network structure and movie attributes on the recommendation performance.

Firstly, we examine the average precision of the two recommendation methods. We can obtain the monthly precision $P(L)$ from May 2004 to December 2004 according to Table 1, and average $P(L)$ for eight months to get the average precision. The average precision of Method I and Method II are given in Fig. 2. Fig. 2 illustrates the influences of w_1 and L on the average precision. It can be seen that if L is fixed, the average precision increases first and then decreases with the increase of w_1. When w_1 is fixed, the average precision also increases first and then decreases with the increase of L. Therefore, there exist optimal values of w_1 and L for the average precision of these two methods. In Fig. 2(a), when

$0.6 \le w_1 \le 0.8$, the average precision is higher. In order to make the average precision as high as possible, we need to consider not only the relation between nodes (i.e., network structure), but also the factors of the node itself (i.e., attributes) in the recommendation process. In Fig. 2(b), if $0.25 \le w_1 \le 0.45$, the average precision is higher. The experiment results show that in order to make the average precision be higher, the optimal values of w_1 are different. When in Method I, we should lay more emphasis on network structure, while in Method II we should focus on attributes. In addition, the effects of Method I and Method II are better than those of the single attribute-based or single structure-based recommendation.



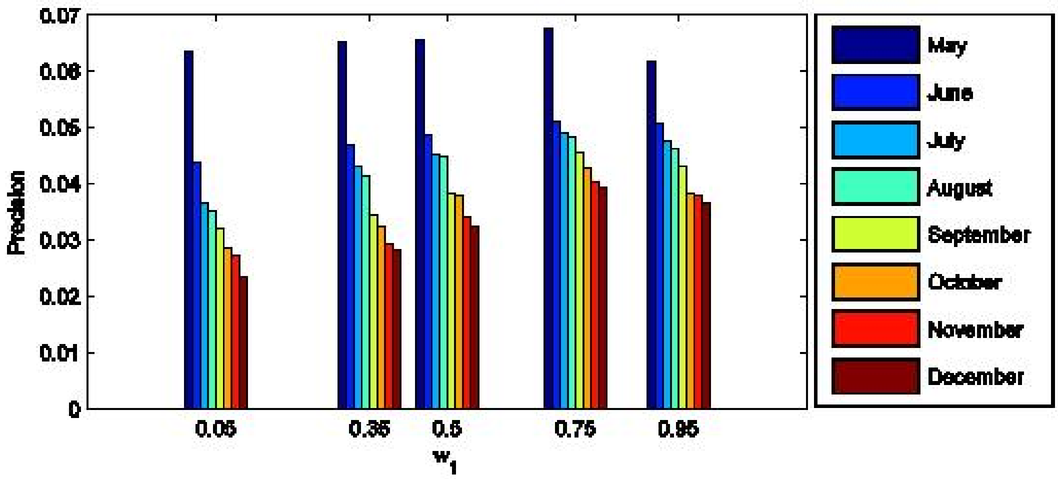
(a) In Method I



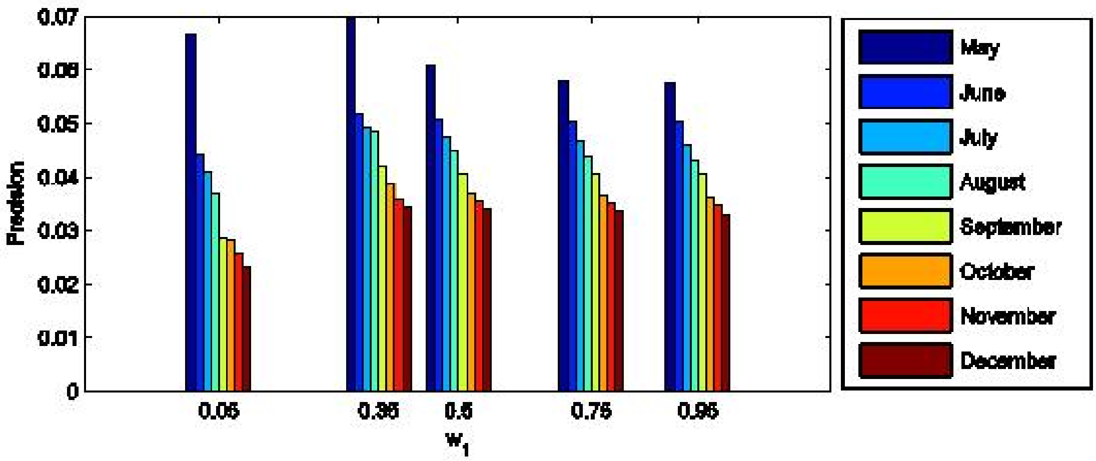
(b) In Method II

Fig. 2. The influences of w_1 and L on the average precision in two methods. X-axis represents the weight of recommendation score which is calculated based on network structure. Y-axis denotes the length of the recommendation list

Next, we evaluate the long-term effects of these two methods in the evolving network. On the one hand, when L is fixed (let L=7), in Fig. 3, we give the precision from May to December of Method I and Method II when w_1=0.05, 0.35, 0.5, 0.75 and 0.95. When w_1 is fixed, the precision is declining from May to December in Fig. 3. In Fig. 3(a), when $w_1 \le 0.75$, the precision for each month will rise with the increase of w_1. If $w_1 > 0.75$, the precision for each month will decrease with the increase of w_1. Hence, the optimal value of w_1 is 0.75for the precision in Method I. However, in Fig. 3(b), when $w_1 = 0.35$, the precision for each month is the highest, that is the optimal value of w_1 is 0.35 in Method II. Additionally, in Fig. 4, we give the change of the precision for each month with the increase of L based on Method I when $w_1 = 0.75$. The change trend of the precision for each month based on Method II is similar to that of Method I with the increase of L. It can be seen that with the increase of L, the precision for each month shows a downward trend. This shows that the length of the recommendation list cannot be too long.

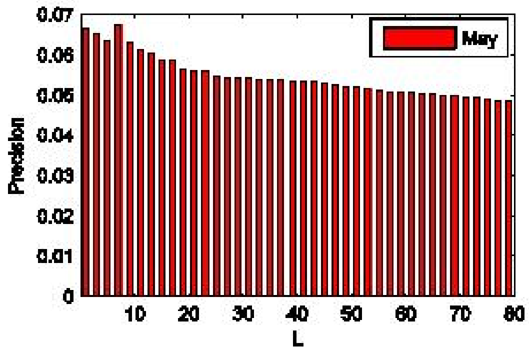


(a) In Method I

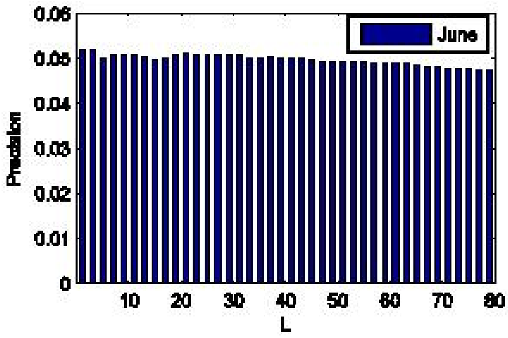


(b) In Method II

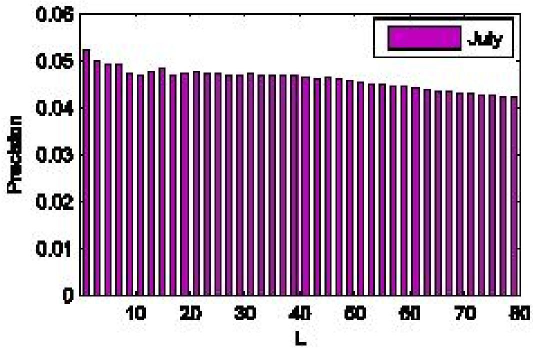
Fig. 3. The influences of different w_1 on the precision for each month when L=7



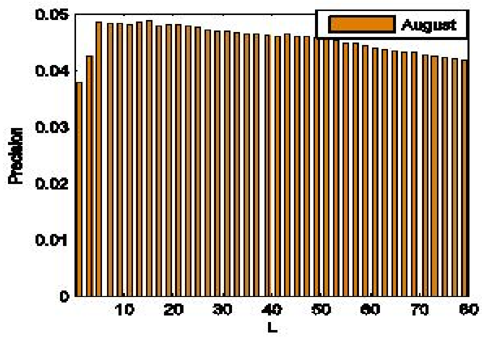
(a) In May



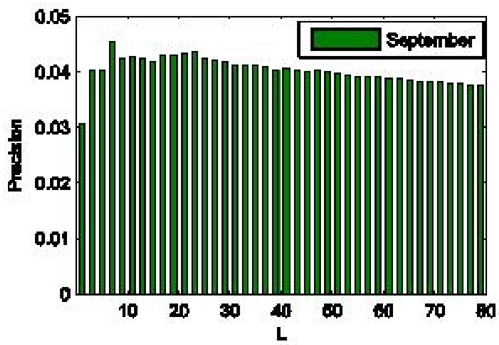
(b) In June



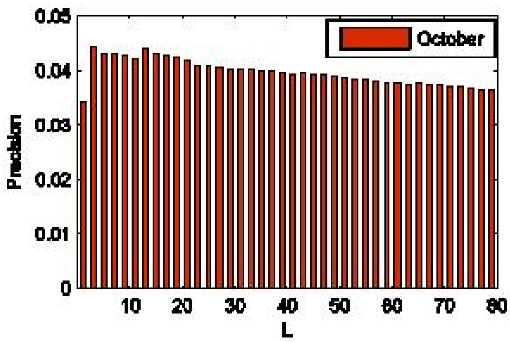
(c) In July



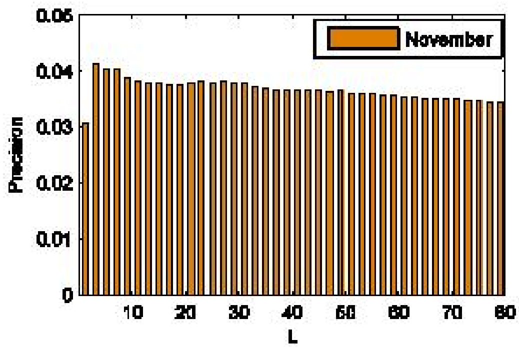
(d) In August



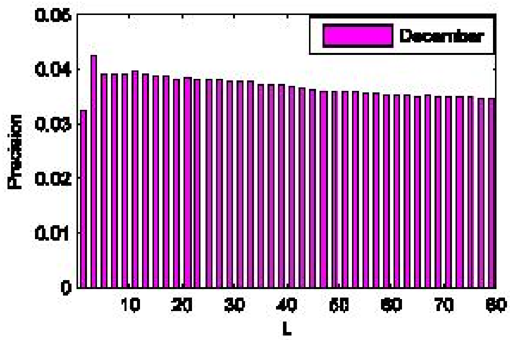
(e) In September



(f) In October



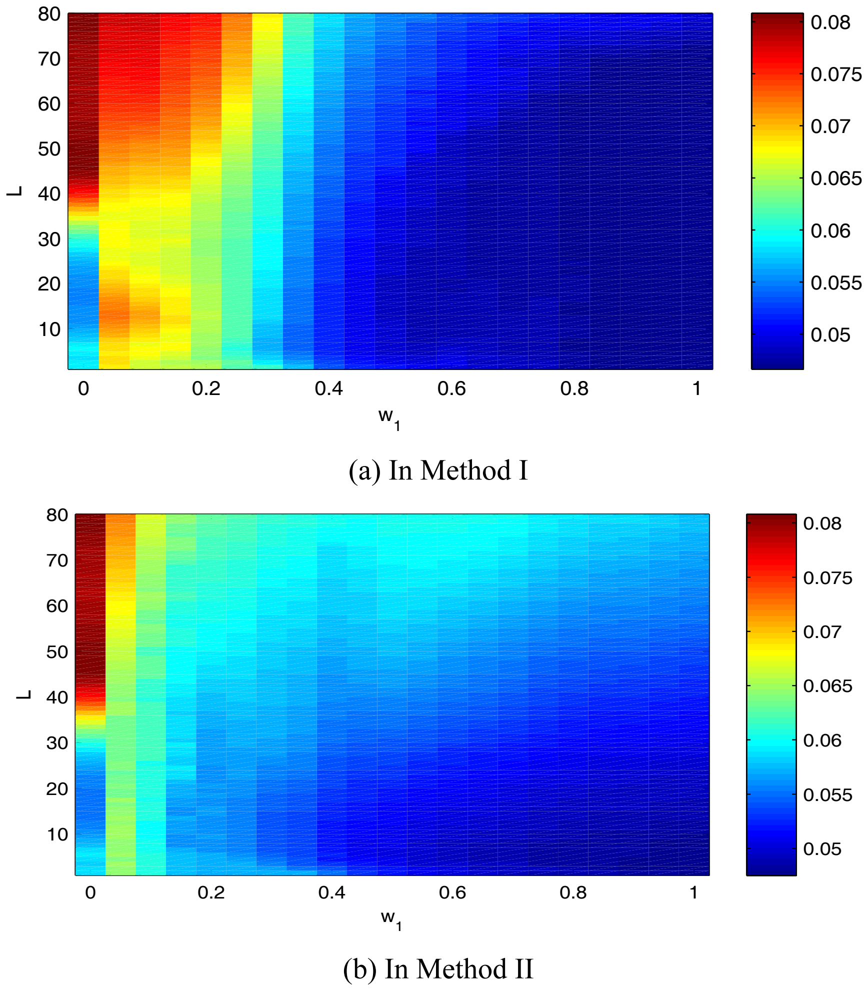
(g) In November



(h) In December

Fig. 4. The influences of L on the precision for each month based on Method I when w_1=0.75

Besides, we focus on the average recommendation novelty of the two methods in evolving networks and get the recommendation novelty $N(L)$ for each month from May 2004 to December 2004 according to Table 2, then average $N(L)$ for eight months to obtain the average recommendation novelty. In Fig. 5(a), we give the average recommendation novelty of Method I in term of w_1 and L. It can be seen that except for $w_1 = 0$, the average recommendation novelty is decreasing with the increase of w_1 when L is fixed. This indicates that the larger w_1 is, the more difficult new movies are recommended. If w_1 is fixed, the average recommendation novelty increases with the increase of L. This shows that new movies are more likely to be recommended when the length of the recommendation list is longer. Fig. 5(b) illustrates the influences of w_1 and L on the average recommendation novelty of Method II. When L is fixed, the average recommendation novelty also decreases with the increase of w_1 except for $w_1 = 0$. Compared with Fig. 5(a), the downtrend of novelty in Fig. 5(b) is not significant. If w_1 is fixed, the average recommendation novelty also increases with the increase of L. Specifically, when $w_1 = 0$, the average recommendation novelty can get the largest value. Comparing Fig. 5(a) and Fig. 5(b), we find that the average recommendation novelty of Method I is larger than that of Method II when $w_1 < 0.4$, while the average recommendation novelty of Method I is smaller than that of Method II when $w_1 \geq 0.4$.



(a) In Method I

(b) In Method II

Fig. 5. The influences of w_1 and L on the average recommendation novelty. The weight w_1 is set as X-axis and the recommendation list length L is set as Y-axis

To further show the performance of proposed methods in this paper, we compare Method I, Method II with other methods in pervious works, respectively, and the results are shown in Table 3 and Table 4. Observing Table 3 and Table 4, we find that for recommendation precision and novelty, the proposed methods in this paper outperform other methods in pervious works.

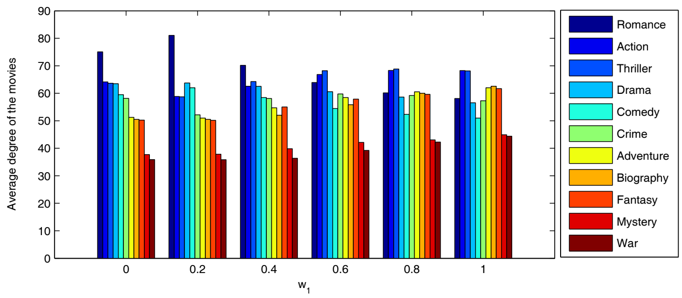
Table 3. The performance of Method I and other methods when L=15

Recommendation methods	*P*	*N*
Cold-start problem, structure-based and attribute-based (Method I)	0.046077	0.071979
Cold-start problem, structure-based	0.043634	0.046637
Cold-start problem, attribute-based	0.031531	0.054985
Structure-based and attribute-based	0.044628	0
Structure-based	0.043061	0
Attribute-based	0.030158	0

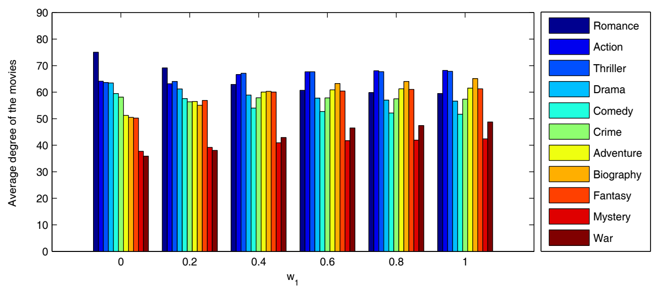
Table 4. The performance of Method II and other methods when L=15

Recommendation methods	*P*	*N*
Cold-start problem, structure-based and attribute-based (Method II)	0.044987	0.064369
Cold-start problem, structure-based	0.042689	0.048779
Cold-start problem, attribute-based	0.031531	0.054985
Structure-based and attribute-based	0.043557	0
Structure-based	0.041563	0
Attribute-based	0.030158	0

Finally, we find that the influences of different w_1 on various genres of movies are different in the two methods (see Fig. 6). From Fig. 6(a), it can be seen that the average degrees of two genres of movies (i.e., Romance and Comedy) increase first and then decrease, while the average degrees first decrease, and then increase for Action, Thriller and Crime. Additionally, the average degrees of certain genres of movies are increasing, such as Adventure, Biography, Fantasy, Mystery and War, while the average degree is decreasing for Drama. Specifically, the average degree for Romance is larger than the average degrees of other genres of movies when w_1=0, 0.2 and 0.4. If w_1=0.6, 0.8 and 1, the average degrees for Action and Thriller are larger. In Fig. 6(b), the average degrees decrease first and then increase for Action and Crime. However, the average degrees of some genres of movies are increasing, including Thriller, Adventure, Biography, Fantasy, Mystery and War, and the average degrees are decreasing for Romance, Drama and Comedy. Comparing Fig. 6(a) and Fig. 6(b), we find that the change trends of the average degrees for Romance, Comedy and Thriller are different. Moreover, in Fig. 6(b), the average degree for War is larger than that for Mystery if w_1=0.4, 0.6, 0.8 and 1, while the average degree for War is smaller than that for Mystery in Fig. 6(a). In order to make the movie degree be larger, we should adopt different recommendation methods for various genres of movies.



(a) In Method I



(b) In Method II

Fig. 6. The average degrees of various genres of movies are given according to different w_1

5 Conclusion

In the context of the evolving user-movie network, two recommendation methods are proposed in this paper. In the two methods, we take into account not only the relation between nodes (i.e., network structure), but also the factors of the node itself (i.e., attributes), and the movie cold-start problem is solved as well. Then we examine the recommendation performance of the two methods by means of two metrics (i.e., precision and novelty), and evaluate the long-term effects of the two methods in the evolving network. We find that the network structure, the movie attributes and the length of the recommendation list affect the recommendation accuracy and the novelty. Firstly, when the average precision of the two methods is higher, the optimal values of their corresponding w_1 are not the same. Results show that when Method I is adopted, we should lay more emphasis on structure-based recommendation, while Method II is adopted, we should focus on attribute-based recommendation. And the effects of Method I and Method II are better than those of the single attribute-based or single structure-based recommendation. By investigating the precision for each month in the network evolution process, we find that the precision for each month is relevant to the weight w_1 and the recommendation list length L. Secondly, the influences of w_1 on the average recommendation novelty of the two methods are not the same. Additionally, in each method, the change trend of the average precision is different from that of the average recommendation novelty with the increase of L. Finally, we find that the average degrees of various genres of movies are also related to w_1. In a word, the performance of Method I and Method II are better than that of other methods in previous works. In addition, the methods proposed in this paper can be applicable to not only the user-movie network, but also the user-video network, the user-book network, and so on. However, these two methods can only solve the movie cold-start problem in the evolving system. In future, we need to study a method which can simultaneously solve the user cold-start problem and the item cold-start problem, and further improve the performance of the recommendation method in the evolving system.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.11331009 and 11601292), overseas returnee research project of Shanxi Province (Grant No. 2015-079).

References

[1] J. Lu, D.-S. Wu, M.-S. Mao, W. Wang, G.-Q. Zhang, Recommender system application developments: a survey, Decision Support Systems 74(2015) 12-32.

[2] F. Ricci, L. Rokach, B. Shapira, Recommender systems: introduction and challenges, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2015, pp. 1-34.

[3] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems 46(2013) 109-132.

[4] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proc. the 10th International Conference on World Wide Web, 2001.

[5] T. Jambor, J. Wang, Optimizing multiple objectives in collaborative filtering, in: Proc. the fourth ACM Conference on Recommender systems, 2010.

[6] M. Khabbaz, L.V.S. Lakshmanan, TopRecs: Top-k algorithms for item-based collaborative filtering, in: Proc. the 14th International Conference on Extending Database Technology, 2011.

[7] C.C. Aggarwal, Recommender Systems, Springer, 2016.

[8] P. Lops, M.-d. Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 73-105.

[9] P.B. Thorat, R.M. Goudar, S. Barve, Survey on collaborative filtering, content-based filtering and hybrid recommendation system, International Journal of Computer Applications 110(2015) 31-36.

[10] R. Burke, Hybrid web recommender systems, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Springer-Verlag Berlin Heidelberg, 2007, pp. 377-408.

[11] F. Yu, A. Zeng, S. Gillard, M. Medo, Network-based recommendation algorithms: A review, Physica A 452(2016) 192-208.

[12] X.-M. Wang, Y. Liu, F. Xiong, Improved personalized recommendation based on a similarity network, Physica A 456(2016) 271-280.

[13] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, Bipartite network projection and personal recommendation, Physical Review E 76(2007) 046115.

[14] L. Lü, M. Medo, C.H. Yeung, Y.-C. Zhang, Z.-K. Zhang, T. Zhou, Recommender systems, Physics Reports 519(2012) 1-49.

[15] S.-J. Zhang, Z. Jin, J. Zhang, The dynamical modeling and simulation analysis of the recommendation on the user-movie network, Physica A 463(2016) 310-319.

[16] D. Roy, A. Kundu, Design of movie recommendation system by means of collaborative filtering, International Journal of Emerging Technology and Advanced Engineering 3(2013) 67-72.

[17] K. Soni, R. Goyal, B. Vadera, S. More, A three way hybrid movie recommendation system, International Journal of Computer Applications 160(9)(2017) 29-32.

[18] T.-G. Hwang, C.-S. Park, J.-H. Hong, S.-K. Kim, An algorithm for movie classification and recommendation using genre correlation, Multimedia Tools and Applications 75(2016) 12843-12858.

[19] Z. Wang, X. Yu, N. Feng, Z. Wang, An improved collaborative movie recommendation system using computational intelligence, Journal of Visual Languages and Computing 25(2014) 667-675.

[20] S. Wei, X. Zheng, D. Chen, C. Chen, A hybrid approach for movie recommendation via tags and ratings, Electronic Commerce Research and Applications 18(2016) 83-94.

[21] C. C. Chen, Y.-H. Wan, M.-C. Chung, Y.-C. Sun, An effective recommendation method for cold start new users using trust and distrust networks, Information Sciences 224(2013) 19-36.

[22] T. Qiu, G. Chen, Z.-K. Zhang, T. Zhou, An item-oriented recommendation algorithm on cold-start problem, EPL 95(5)(2011) 58003.

[23] B. Lika, K. Kolomvatsos, S. Hadjiefthymiades, Facing the cold start problem in recommender systems, Expert Systems with Applications 41(2014) 2065-2073.

[24] J.-H. Liu, T. Zhou, Z.-K. Zhang, Z. Yang, C. Liu, W.-M. Li, Promoting cold-start items in recommender systems, PLoS ONE 9(12)(2014) e113457.

[25] D.-D. Zhao, A. Zeng, M.-S. Shang, J. Gao, Long-term effects of recommendation on the evolution of online systems, Chinese Physics Letters 30(11)(2013) 118901.

[26] X. Hu, A. Zeng, M.-S. Shang, Recommendation in evolving online networks, The European Physical Journal B 89(2)(2016) 46.