Eur. Phys. J. B (2016) 89: 46
DOI: 10.1140/epjb/e2016-60509-9

THE EUROPEAN PHYSICAL JOURNAL B

Regular Article

Recommendation in evolving online networks

Xiao Hu[1], An Zeng[2,a], and Ming-Sheng Shang[3]

[1] Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, P.R. China
[2] School of Systems Science, Beijing Normal University, Beijing 100875, P.R. China
[3] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, P.R. China

Received 26 June 2015 / Received in final form 20 December 2015
Published online 17 February 2016 –

Abstract. Recommender system is an effective tool to find the most relevant information for online users. By analyzing the historical selection records of users, recommender system predicts the most likely future links in the user-item network and accordingly constructs a personalized recommendation list for each user. So far, the recommendation process is mostly investigated in static user-item networks. In this paper, we propose a model which allows us to examine the performance of the state-of-the-art recommendation algorithms in evolving networks. We find that the recommendation accuracy in general decreases with time if the evolution of the online network fully depends on the recommendation. Interestingly, some randomness in users' choice can significantly improve the long-term accuracy of the recommendation algorithm. When a hybrid recommendation algorithm is applied, we find that the optimal parameter gradually shifts towards the diversity-favoring recommendation algorithm, indicating that recommendation diversity is essential to keep a high long-term recommendation accuracy. Finally, we confirm our conclusions by studying the recommendation on networks with the real evolution data.

1 Introduction

Due to the rapid expansion of the Internet, there are a huge amount of online resources nowadays [1]. Therefore, online users need to process much more information before they can reach what they really want. With great brilliance, various kinds of information filtering tools were investigated to address this information overload problem [2,3]. In particular, the recommender system is a promising way which provides users with personalized suggestions of the most relevant items by analyzing users' activity records and available personal profiles [4]. This technology has been widely applied in many fields, such as search engine, e-commerce systems, mobile Internet and so on.

In recent years, a variety of recommendation algorithms have been proposed. For example, one of the most successful methods is the collaborative filtering (CF) with user-based and item-based versions [5–7]. The matrix factorization algorithms have also been widely investigated [8,9]. The recommender system attracts attention from physicists as well. They study the recommendation process based on the user-item bipartite graph, and develop several efficient algorithms by employing some well-known physical processes such as mass diffusion (MD) [10] and heat conduction (HC) [11]. A remarkable finding is that the hybridization of MD and HC can solve the diversity-accuracy dilemma in recommendation [12]. These methods lead to many extensions, such as the preferential diffusion [13], the similarity-preferential diffusion [14], and network manipulation [15]. In addition, the long-term influence of recommendation on global diversification has been studied lately [16,17].

However, the existing works only focus on the performance of the recommendation algorithms in a single round of recommendation [18]. This actually only reflects the short-term recommendation accuracy of the algorithm. As the online networks are evolving all the time [19–22], one has to investigate the long-term accuracy of the algorithms, i.e. what is the recommendation accuracy after the recommendation algorithm guides the evolution of the online network for a long time. This is not a trivial problem because maximizing the short-term accuracy cannot guarantee high long-term accuracy. It has already been pointed out that many recommendation algorithms with high short-term accuracy tend to recommend most popular items [16,17]. If the online network evolves with such recommendation algorithms, gradually only a small number of most popular items will attract most of the links. The recommendation accuracy would be seriously lowered under this network structure. However, which algorithms will lead to this effect and how strong this effect is on recommendation accuracy still remain unclear. Therefore, it is necessary to explore the performance of recommendation algorithms in the evolving networks.

In this paper, we introduce a model to examine the long-term performance of the state-of-the-art recommendation algorithms in evolving networks. In particular,

[a] e-mail: an.zeng@unifr.ch

we couple the evolution of the online network with the recommender systems by assuming that users accept the recommendation generated by the algorithms. We find that the recommendation accuracy in general decreases with time if the evolution of the online network fully depends on the recommendation. Moreover, when a hybrid recommendation algorithm is applied, we find that the optimal parameter gradually shift towards the diversity-favoring recommendation algorithm, indicating that recommendation diversity is essential to keep a high long-term recommendation accuracy. Finally, we study the recommendation on networks from real evolution data and observe similar results.

2 Recommendation algorithms

We first introduce five simple and representative recommendation algorithms that we will consider in this paper. They are user-based collaborative filtering (UCF), item-based collaborative filtering (ICF), mass diffusion (MD), heat conduction (HC) and the hybridization of these two algorithms (Hybrid). In general, all these methods are personalized and uncover users potential interests based on their individual historical actions.

A recommender system can be represented as a bipartite network $G \equiv (U, O, E)$, where $U = \{u_1, u_2, \ldots, u_m\}$, $O = \{o_1, o_2, \ldots, o_n\}$ and $E = \{e_1, e_2, \ldots, e_k\}$ are the sets of users, items and edges, respectively. The bipartite network is presented by an adjacency matrix A, where the element $a_{i\alpha} = 1$ if user i has collected item α, and otherwise $a_{i\alpha} = 0$ (throughout this paper we use Greek and Latin letters, respectively, for item- and user-related indices).

Collaborative filtering is one of the most famous recommendation methods. It consists of user-based collaborative filtering version and item-based collaborating filtering version [23,24]. The user for whom the recommendation is done is called target user. In UCF, the recommendation scores of uncollected items for target user i is calculated through his/her similarity to the users who have collected the same item with i. The final recommendation score for each item α to user i can be computed as:

$$P_{i\alpha} = \sum_{j=1}^{m} s\left(i, j\right) a_{i\alpha} \tag{1}$$

where $s\left(i, j\right)$ denotes the similarity between user i and j. The measure of similarity is subject to definition. Here we use the Slaton index measure in the bipartite network [25]. Denote $\Gamma\left(i\right)$ as the neighbor set of user i and k_i as the degree of user i, the similarity between node i and j can be expressed as:

$$s\left(i, j\right) = \frac{\left|\Gamma\left(i\right) \cap \Gamma\left(j\right)\right|}{\sqrt{k_i k_j}}. \tag{2}$$

The resulting recommendation list of uncollected items for the target user i is then sorted according to $P_{i\alpha}$ in descending order. In ICF, the recommendation score of an item α for the target user i is calculated by its similarity to the collected items of i, i.e.,

$$P_{i\alpha} = \sum_{\beta=1}^{n} s\left(\alpha, \beta\right) a_{i\beta}. \tag{3}$$

Like UCF, the similarities between items are again computed with the Slaton index [25].

Unlike the collaborative filtering algorithm, the diffusion-based methods work on the bipartite network through the transformation of resources. For a target user i, the diffusion process starts by assigning one unit of resource to each item collected by user i, then redistributes the resource through the user-item bipartite network. The final resource of item α received from the user i can be calculated as:

$$P_{i\alpha} = \sum_{\beta=1}^{n} w_{\alpha\beta} a_{i\beta} \tag{4}$$

where $w_{\alpha\beta}$ is the element of the diffusion matrix W. By denoting $k_\beta = \sum_{l=1}^{N} a_{l\beta}$ and $k_j = \sum_{\alpha=1}^{M} a_{j\alpha}$ as the degree of item β and user j respectively, the diffusion matrix can be written as:

$$W_{\alpha\beta} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^{\lambda}} \sum_{j=1}^{N} \frac{a_{j\alpha} a_{j\beta}}{k_j} \tag{5}$$

where the parameter λ adjusts the relative weight between the MD and HC, and it decreases from 0 to 1. $\lambda = 0$ gives us the pure HC algorithm [11], and $\lambda = 1$ gives us the pure MD algorithm [10]. The resulting recommendation list of uncollected items is stored according to $P_{i\alpha}$ in descending order. Such hybrid algorithm can generate both accurate and diverse recommendation, especially when λ is tuned to the optimal value [12].

3 Methods

3.1 Real data

In this paper, we consider three real data sets, i.e. MovieLens, Netflix and Delicious. Among these three data sets, MovieLens and Netflix are similar. Both of them are users' ratings on movies and the rating scales are from one (i.e. worst) to five (i.e. best). These two data sets are relatively dense. The main difference between MovieLens and Netflix is that MovieLens data is collected by a research project while Netflix data is collected from a real online commercial website. Unlike MovieLens and Netflix, Delicious is a data set about users collection of web bookmark. This data is unweighted and much sparser than MovieLens and Netflix. If a user has collected a bookmark on this website, a link is constructed from this user and the bookmark.

As we aim to investigate the recommendation performance when the evolution of the bipartite networks is guided by the recommendation algorithms, we eliminate

Table 1. The description of the data. the sparsity is $\frac{links}{users \cdot items}$.

Data set	Users	Items	Links	Sparsity
MovieLens	943	1682	82 520	5.20×10^{-2}
Netflix	2294	1861	71 074	1.70×10^{-2}
Delicious	4733	3058	41 924	2.90×10^{-3}

all the negative ratings and use only the recommendation algorithms for unweighted networks for simplicity. Here we only consider all ratings higher than 2 as links. After coarse gaining from the original MovieLens data, which contains 10^5 ratings, the rest data has 82 520 links in total. In the same way, the subset of Netflix consists of 2294 users and 1861 items, and this set contains 71 074 links. Finally, after randomly selecting 5000 users and 5000 bookmarks, there are finally 4733 non-isolated users and 3058 non-isolated items with 41 924 links in total. In addition, we have checked that in MovieLens there are only 17% negative ratings and in Netflix there are only 15% negative ratings. Therefore, the main information is contained in the positive ratings. The basic statistics of these data sets are reported in Table 1.

3.2 Network evolution model

In order to study the long-term performance of the recommendation algorithms, we consider a model in which the evolution of online networks is driven by recommendation algorithms (RA). In the model, we assume that users would accept the recommendations by randomly selecting one of L highest ranked uncollected items from the recommendation list, resulting in adding new links between users and items in the bipartite network. For comparison, we also consider the case where the evolution of the network is driven by preferential attachment (PA).

In practice, the real network is divided into a training set E^T and a probe set E^P. At each step, a random user is selected as the active user, and the recommendation scores of all items for him/her are then evaluated for based on the training set E^T. In one *macro-step* of our simulation, we randomly choose 10% of users to be active. We assume each active user would accept the recommendations by randomly selecting one of L uncollected item with the highest score, i.e. we add a link between the active user and this item in the training set. After this, a new training set will be obtained E_n^T. The next macro-step of network evolution will be based on E_n^T. In each macro-step, the probe set E^P will be used to examine the recommendation accuracy of this *macro-step* (see description below). Note that, when the network evolution is driven by PA, E_n^T is obtained by assuming at each macro-step the active users will randomly select an item with the probability proportional to its degree. The recommendation algorithm in this case will only be used to measure the recommendation accuracy with E_n^T and E^P at each macro-step.

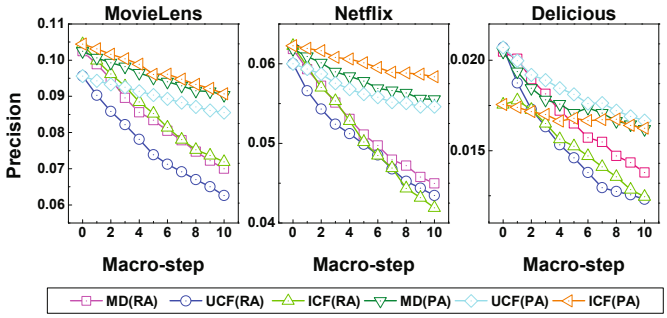

Fig. 1. The evolution of the *precision*. The real data is divided into 90% E^T and 10% E^P data which are respectively used as the initial configuration for evolution and for measuring the recommendation accuracy.

3.3 Recommendation performance

Recommendation accuracy is the most important aspect in evaluating the recommendation performance. In real systems, users usually consider only the top part of the recommendation list and they rarely go far down the list. Thus, we adopt the accuracy measure called *precision*. For a target user i, the recommendation list is first generated by running recommendation algorithms on E_n^T and the *precision* of the recommendation is defined as:

$$P_i(L) = \frac{d_i(L)}{L} \tag{6}$$

where $d_i(L)$ represents the number of i's probe set items in the top-L places in the recommendation list (we set $L = 20$). By averaging $P_i(L)$ for all the users, we obtain the precision $P(L)$ for the whole system.

4 Results

4.1 The evolution of recommendation accuracy

Our analysis focuses on the dependence of recommendation accuracy on macro-steps. We first consider the mass diffusion, user-based and item-based collaborative filtering algorithms. The results are shown in Figure 1. When the network evolution is driven by RA (i.e. users create new links only by selecting items from their recommendation lists), the precision of all three recommendation algorithms tend to decrease. This indicates that if users only rely on recommendation algorithms, in the long term they will find more and more irrelevant items in the recommendation lists. The quantitative results are reported in Table 2. For comparison, we also report the precision of these three recommendation algorithms when the network evolution is driven only by PA (i.e. users choose items with the probability proportional to items' degree). As shown in Figure 1, the decay of precision in this case is much slower than that in the recommendation-driven evolution.

Generally, recommender systems only show users a list of most relevant items. In real online systems, the length of recommendation list is usually rather short, roughly

Table 2. The comparison of the short-term *precision* and long-term *precision*. MS@T denotes the Tth macro-step. For the Hybrid method, λ is set as the optimal value at $T = 0$. For Hybrid* method, λ is the optimal value at each macro-step (λ^*). The results are obtained by averaging 100 independent realizations. The standard deviation is not shown because it is very small (i.e. around 10^{-4}).

MovieLens						
MS@T	UCF	ICF	Mass	Heat	Hybrid	Hybrid*
MS@0	0.096	0.104	0.103	0.007	0.122	0.122
MS@9	0.065	0.074	0.072	0.004	0.087	0.090
Netflix						
MS@T	UCF	ICF	Mass	Heat	Hybrid	Hybrid*
MS@0	0.060	0.062	0.062	0.001	0.0690	0.0690
MS@9	0.044	0.043	0.046	0.004	0.0473	0.0478
Delicious						
MS@T	UCF	ICF	Mass	Heat	Hybrid	Hybrid*
MS@0	0.021	0.018	0.020	0.000	0.0211	0.0211
MS@9	0.013	0.013	0.014	0.000	0.0138	0.0140

around 20. In usual online systems such as Netflix, Amazon and Delicious, there are thousands of movies, millions of products and billions of web pages. Presenting only top most highly ranked items to users will remove the attention to other more than 99% items. If the recommendation algorithms tend to give popular items with high score (like MD, UCF), then these top-20 items will be mostly popular items. This is not a serious problem for one round of recommendation. However, if this recursively happens (i.e. each time users only select the most popular items as they can only see these popular items in their top-20 lists), the whole market will be dominated by a very small number of items. In network language, it is a small number of items attracting all the links from users. Once this structure is formed, recommendation algorithms can no longer find any relevant unselected items for users, causing very low recommendation accuracy.

The network evolution based on preferential attachment, on the other hand, is much less harmful. It assumes that users select items with the probability proportional to items' cumulative degree. Even though the preferential attachment is biased to the large degree items, the small degree items will not be completely ignored. In the long-term evolution, there will be some hubs in the network, but other items will still have some links connecting to them. Thanks to these a few connections, the small degree items will still be recommended (i.e. appear in users' recommendation list) since they can be reached by the diffusion when the diffusion-based recommendation algorithms are used, and they can have nonzero similarity to other items when similarity-based recommendation algorithms are used. Therefore, the long term recommendation accuracy is higher when the network evolution is guided by preferential attachment than recommendation.

Preferential attachment totally ignores the network structure and the user interests that the structure captures. This for sure will reflect in the single-round recommendation accuracy (i.e. the recommendation accuracy is lower when preferential attachment is used to generate the recommendation list rather than the recommendation algorithms). However, this drawback is secondary when the user-item network is evolving. The unhealthy item degree distribution that dominated by a small number of items will ruin the recommendation accuracy more seriously. Taken together, the long term diversification of the item degree distribution is the key to keep high recommendation performance in the long term. That is to say, users should not only select items from the recommendation lists, but also have some probability to select small degree items outside the recommendation lists. This randomness in item selection will lead to healthier item degree distribution for the long-term future recommendation.

The reinforcing effect of the recommender systems shown in Figure 1 exhibits how the system will be like in the long term if users overly rely on the recommendation. This effect has to be confirmed empirically. In principle, one can roughly check the evolution of the recommendation accuracy in real data by counting the fraction of disliked links. However, as the number of disliked links is strongly affected by many factors, such as the number of users following recommendation and the quality of the new movies, a more direct empirical measure remains to be designed.

4.2 The effect of recommendation list length

Given a recommendation algorithm, one effective way to increases users' selection diversity is to increase the recommendation list length. In this way, users will have more items to choose. However, if the recommendation list length is set to be too long, many irrelevant items will be included in the recommendation lists, which decreases users' satisfaction. In Figure 2, we investigate the effect of recommendation list length on the long-term recommendation accuracy. As expected, the recommendation accuracy indeed increases with the recommendation list length at the beginning. However, after a certain value, further increasing the list length would only decrease the recommendation accuracy. This is because many users are guided to select some irrelevant items from the recommendation lists, making the resultant network contain less valuable information about users' real interest. Moreover, we observe that the optimal length increases with

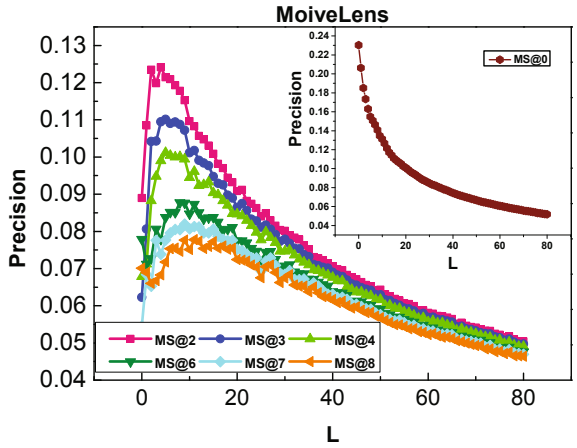

Fig. 2. The long term precision with different recommendation list lengths. MS@T denotes the Tth macro-step. The inset shows the correlation between the recommendation list and *precision* at $T = 0$. The recommendation algorithm is MD and the initial network is MovieLens.

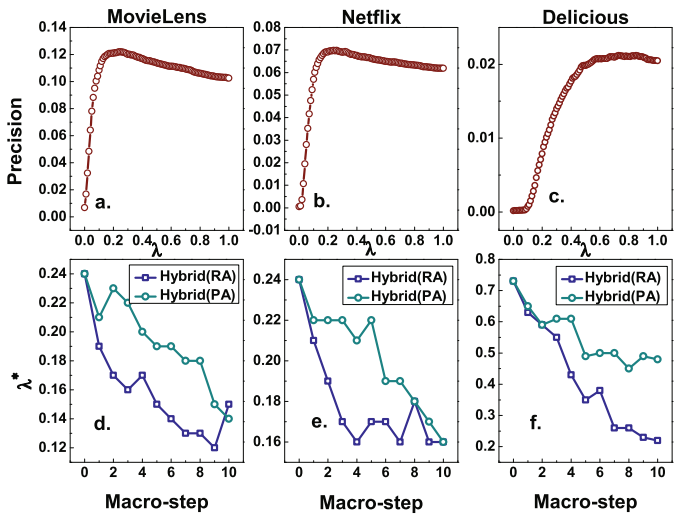

Fig. 3. The evolution of the *precision* when the Hybrid method is used. The upper subplots are the correlation between λ and *precision* at macro-step = 0, and the bottom subplots are the evolution of the λ^*.

macro-steps. This indicates that in order to keep high recommendation accuracy in longer future, more diverse user selections are needed.

4.3 The evolution of hybrid recommendations

We further study the long-term performance of the well-known hybrid method in reference [12]. As shown in Figure 3, the hybridization of MD and HC can achieve higher recommendation accuracy than either individual method. In particular, when tuning the hybrid parameter λ, the precision can reach an optimal value. Here, we focus on how the optimal λ^* changes with macro-steps. In Figures 3d–3f, one can see that the λ^* generally decreases with macro-steps, indicating that more weight should be given to the heat conduction algorithm. This is natural as the heat conduction algorithm is specialized in recommending niche but relevant items to users. This result confirms again that recommendation diversity is essential to keep a high long-term recommendation accuracy. In Table 2, we compare the *precision* of the hybrid method with a fixed λ^* (obtained by maximizing precision at $macro - step = 0$) and the hybrid method with a dynamical λ^* (obtained by maximizing precision at each macro-step). The precision of the hybrid method with a dynamical λ^* is slightly higher.

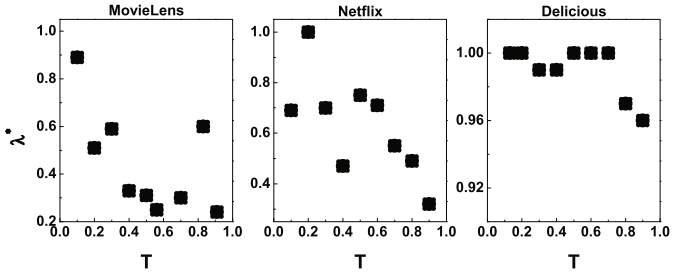

Fig. 4. The evolution of λ^* estimated with the real network evolution data.

We finally investigate the evolution of λ^* with the real network evolution data. Here, we use three real network data sets: MovieLens, Netflix and Delicious. In all these three online web sites, recommender systems are implemented either directly or indirectly. Therefore, the real evolution of these three user-item networks can be assumed to be somehow influenced by recommendations. To see how λ^* changes with time in these three real networks, we divide the data set according to the real time stamps on links. The time stamps are first normalized in a way that the maximum time stamp is equal to 1. Given a testing time t ($t < 1$), all the links created earlier than t are placed in the training set and 10% of the links created later than t are assigned to the probe set. By increasing t, we can examine the optimal λ^* at different real evolution time. The results on three real networks are presented in Figure 4. Interestingly, consistent with the results in the network evolution model, λ^* also tend to decrease with t in real data. This indicates that the hybrid method indeed needs to shift to the diversity-favoring recommendation algorithm for long-term accuracy.

Even though the general downwards trend of λ^* is consistent in Figures 3 and 4, there is some magnitude difference for the same data set in these two figures. There are actually many factors causing this phenomenon. The main possible factor is that the way users select items is different in the artificial model and real data. In Figure 3, we assume that all the users select the items from the recommendation list. While in Figure 4, the users select items according to the real data. This means that in Figure 4 the users are not for sure selecting the relevant items according to what the recommendation algorithms show to them in Figure 3. In real data, users may discover new items (items with zero similarity to what the users selected before or items with zero degree). If this happens often, the difference between simulation and real data will

be big. The opposite magnitude in Figures 3 and 4 implies that the fraction of users following recommendation is very different. Netflix is a web site relying significantly on recommendation, so more users follow recommendation on Netflix. Therefore, the decreasing magnitude of λ^* is large in real Netflix data. The Delicious web site, however, is influenced less by recommendation, so fewer users follow recommendation on Delicious, resulting in smaller decreasing magnitude of λ^* in real Delicious data.

5 Conclusions and discussions

Recommender systems are widely developed to effectively support users' decision-making process in the online systems. Though many recommendation algorithms have been proposed in the literature, the performance of these algorithms in evolving networks still remains unclear. In this paper, we develop a model to study the performance of state-of-the-art recommendation algorithms in evolving networks. We find that even though the recommendation algorithms are very effective in helping users to find the relevant items in the short term, its prediction accuracy keeps decreasing in the long term if the evolution of the online network fully depends on recommendation. Interestingly, some randomness in users' choices can significantly improve the long-term accuracy of the recommendation algorithm. We apply the hybrid recommendation algorithm in the evolving network and optimize its parameter in each time step. We find that the optimal parameter gradually shift towards the diversity-favoring recommendation algorithm, indicating that recommendation diversity is essential to keep a high long-term recommendation accuracy. We finally confirm this phenomenon by studying the recommendation on networks from real evolution data.

The model developed in this paper, though simple, captures several theoretically important features of recommendation in evolving networks, such as the reinforcing effect and the nonlinear effect of the parameters. There are a number of interesting extensions that could be done in the near future. For instance, in reality only some users would follow recommendation while the rest of them choose items on their own. One can use a parameter p to control the fraction of users following recommendation and study the influence of p on the long-term recommendation performance. Nevertheless, it is very hard to model users' behavior when they are not guided by recommendation as users' behavior is determined by many factors such as age, gender, job, location, etc. A more realistic model for the evolution of online systems remains to be designed. Moreover, the item acceptance by users could be modeled in a more realistic way. For example, the concept drift effect [19] could be taken into account. In addition, the effect of negative ratings on the recommendation results [26] could be an important role during the evolution. Therefore, how to incorporating these realistic factors in the evolution model will be our research focus in the near future.

Finally, the evolution of the user-item bipartite network has been shown to be strongly influenced by users' social network. Especially, the users' social credit/trust is a very important issue for recommendation algorithms and it can significantly enhance the recommendation accuracy. Analyzing the social trust in the evolution will improve the quality of the recommendation system [27,28]. Thus, studying the recommendation algorithms with the co-evolution of users' social network and user-item bipartite network would be of great interest for both physicists and computer scientists [29–32].

We acknowledge Prof. Yi-Cheng Zhang for helpful discussion. This work is supported by the National Natural Science Foundation of China under Grant Nos. 61370150, 61433014 and the Sci & Tech Research Project of Chongqing (cstc2015jcyjB0244). A.Z. acknowledges the support from the Youth Scholars Program of Beijing Normal University (Grant No. 2014NT38).

References

1. A. Broder, R. Kumar, F. Moghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Comput. Netw. **33**, 309 (2000)
2. G. Adomavicius, A. Tuzhilin, IEEE Trans. Knowl. Data Eng. **17**, 734 (2005)
3. F. Cacheda, V. Carneiro, D. Fernandez, V. Formoso, ACM Trans. Web. **5**, 1 (2011)
4. L. Lü, M. Medo, C.H. Yeung, Y.C. Zhang, Z.K. Zhang, T. Zhou, Phys. Rep. **1**, 519 (2012)
5. J. Wang, P.V. Arjen, J.R. Marcel, in *Proceedings of ACM SIGIR conference on Research and Development in Information Retrieval* (2006), pp. 501−508
6. J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, Commun. ACM **40**, 77 (1997)
7. J.L. Herlocker, J.A. Konstan, K. Terveen, J.T. Riedl, ACM Trans. Inf. Syst. Secur. **22**, 5 (2004)
8. Y. Koren, R. Bell, C. Volinsky, Computer **42**, 30 (2009)
9. Y. Hu, Y. Koren, C. Volinsky, in *IEEE International Conference on Data Mining, 2008*, pp. 263–272
10. T. Zhou, J. Ren, M. Medo, Y.C. Zhang, Phys. Rev. E **76**, 046115 (2007)
11. Y.C. Zhang, M. Blattner, Y.K. Yu, Phys. Rev. Lett. **99**, 154301 (2007)
12. T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Proc. Natl. Acad. Sci. USA **107**, 4511 (2010)
13. L.Y. Lü, W.P. Liu, Phys. Rev. E **83**, 066119 (2011)
14. A. Zeng, V. Alexandre, M. Medo, Y.C. Zhang, ACM Trans. Inf. Syst. Secur. **22**, 5 (2004)
15. F.G. Zhang, A. Zeng, Europhys. Lett. **100**, 58005 (2012)
16. A. Zeng, C.H. Yeung, M.S. Shang, Y.C. Zhang, Europhys. Lett. **97**, 18005 (2012)
17. D.D. Zhao, A. Zeng, M.S. Shang, J. Gao, Chin. Phys. Lett. **30**, 118901 (2013)
18. P. Cremonesi, Y. Koren, R. Turrin, in *Proceedings of ACM conference on Recommender Systems* (2010), pp. 39−46
19. Y. Koren, Commun. ACM **53**, 89 (2010)
20. C. Aggarwal, S. Karthik, ACM Comput. Sur. **47**, 10 (2014)
21. J. McAuley, L. Jure, in *Proceedings of the international conference on World Wide Web* (2013), pp. 897−908

22. C. Danescu-Niculescu-Mizil, W. Robert, J. Dan, L. Jure, P. Christopher, in *Proceedings of the international conference on World Wide Web* (2013), pp. 307−318
23. K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, Y. Yu, in *Proceedings of ACM SIGIR conference on Research and development in Information Retrieval* (2012), pp. 661−670
24. B. Xu, B. Jiaj, C. Chun, C. Deng, in *Proceedings of the 21st international conference on World Wide Web* (2012), pp. 21−30
25. G. Salton, M.J. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, Auckland, 1983)
26. W. Zeng, Y.X. Zhu, L. L, T. Zhou, Physica A **390**, 4486 (2011)
27. P. Massa, P. Avesani, in *Proceedings of ACM conference on Recommender Systems* (2007), pp. 17−24
28. B. Liu, Zh. Yuan, in *Proceedings of the workshop on Context-Aware Movie Recommendation* (2010), pp. 53−56
29. H. Ma, D. Zhou, C. Liu, M. Lyu, I. King, in *Proceedings of ACM international conference on Web Search and Data Mining* (2011), pp. 287−296
30. M. Jamali, M. Ester, in *Proceedings of ACM conference on Recommender System* (2010), pp. 135−142
31. W. Zeng, L. Chen, in *Proceedings of the Annual ACM Symposium on Applied Computing* (2013), pp. 237−244
32. L. Chen, W. Zeng, Q. Yuan, Exp. Syst. Appl. **40**, 2889 (2013)