Physica A 467 (2017) 490–498

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Long-term effects of user preference-oriented recommendation method on the evolution of online system

Xiaoyu Shi [a], Ming-Sheng Shang [a,*], Xin Luo [a], Abbas Khushnood [b], Jian Li [b]

[a] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, 400714, China

[b] School of Computer Science and Engineer, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China

HIGHLIGHTS

- With consideration of heterogeneity of real user, we propose a novel personalized recommendation method based on the user preference.
- Our main focus is on evaluating the health state of ecosystem in the long-term evolution by building an evolution model, which simulates the mutual feedback between user choices and recommender system.
- We find that there is a good trade-off between short- and long-term performances of recommendation if online system allows multiple recommenders work simultaneously and assigns the optimal recommender to each user in the individual level.

ARTICLE INFO

Article history:
Received 18 May 2016
Received in revised form 1 September 2016
Available online 12 October 2016

Keywords:
Recommender system
User preference-oriented
Personalized recommender method
Evolution
Diversity

ABSTRACT

As the explosion growth of Internet economy, recommender system has become an important technology to solve the problem of information overload. However, recommenders are not one-size-fits-all, different recommenders have different virtues, making them be suitable for different users. In this paper, we propose a novel personalized recommender based on user preferences, which allows multiple recommenders to exist in E-commerce system simultaneously. We find that output of a recommender to each user is quite different when using different recommenders, the recommendation accuracy can be significantly improved if each user is assigned with his/her optimal personalized recommender. Furthermore, different from previous works focusing on short-term effects on recommender, we also evaluate the long-term effect of the proposed method by modeling the evolution of mutual feedback between user and online system. Finally, compared with single recommender running on the online system, the proposed method can improve the accuracy of recommendation significantly and get better trade-offs between short- and long-term performances of recommendation.

1. Introduction

The last decade has witnessed an explosion of development in the area of Internet economy: there are ten thousands of movies and books, and billions of web pages. As a result, it forces us to live in an information overload society: the amount of information, especially on Internet, is increasing far quickly than our ability to process it. In this context, most popular web sites such as Amazon, Netflix and YouTube provide the recommender system, which attempts to facilitate user navigation

* Corresponding author.
E-mail address: msshang@cigit.ac.cn (M.-S. Shang).

http://dx.doi.org/10.1016/j.physa.2016.10.033

by suggesting user new possibly relevant items and thus increases customers satisfaction and their profits [1,2]. Namely, recommender systems seek to predict users' non-considered preference typically according to their historical activities. Until now, numerous recommenders based on different ideas and concepts have been proposed, including collaborative filtering [3–5], the content-based method [6], spectral analysis [7,8], latent semantic models and Dirichlet allocation [9,10], iterative self-consistent refinement [11,12] as well as topology adaptation method [13].

Although recommenders have been greatly developed, there still are a lot of rooms for improvement. The traditional method supports the recommendation services for all users by adopting single recommender and always ignores the heterogeneity between users. It is well-understood that different recommenders, even ones that have quantitatively similar behavior on common accuracy metrics, produce recommendations that differ in ways that users can perceive and that may impact their ability to meet different user needs, or the needs of different users [14]. One way of making use of the advantage of different algorithms to improve recommendation is through hybrid recommender. For example, hybrid method combining heat conduction and probability spreading algorithm (HHP) is proposed to achieve better recommendation performance [15]. With a tunable hybrid parameter, the HHP method provides a smooth yet non-trivial transition from one method to the other. Besides, each real system is shown to have its own optimal hybrid parameter. Guan et al. propose UHHP, a advanced HHP, that apply HHP in individual level, namely each user has his/her own personalized hybrid parameter to adjust [16]. In order to improve higher accuracy than standard collaborative filtering (CF), Liu et al. [17] introduce a modified collaborative filtering algorithm by using the second-order correlations. Zhu et al. [18] adopt mutual correction of forward and backward similarity estimations to design personalized recommender. By considering the users' similarity direction and the second-order correlations to depress the influence of mainstream preferences, Guo et al. [19] propose the directed second-order CF (HDCF) algorithm specifically to address the challenge of accuracy and diversity of the CF algorithm. Furthermore, for solving the accuracy–diversity dilemma, the directed random walks method was proposed [20]. In the aspect of user heterogeneity, based on a weighted projection of the user–object bipartite network, it studies the effects of user tastes on the mass-diffusion-based personalized recommendation algorithm [21]. Furthermore, Shang et al. [22] analysis the structure and evolution of web-based user–object networks in an empirical way. In addition, those existing methods put considerable attentions on short-term effects such as recommendation accuracy and user privacy, the long-term mutual feedback between user and recommender system has been neglected so far. Zeng et al. [23] propose a model of network evolution to study the complex dynamics induced by this feedback, and remark that item diversity can be enhanced by sacrificing a small fraction of recommendation's short-term accuracy in exchange for higher long-term diversity. Zhao et al. [24] also analysis the long-term effects of different recommenders on the evolution of online systems.

In this paper, we explore another approach to solve the personalized recommendation problem. Based on the heterogeneity of real users, we propose a novel personalized recommendation method running on the individual level, where each user can have his/her special recommender. The designed recommender system supports more than one recommender simultaneously. Meanwhile, with consideration of the recommendation results affect the growth of the user–item network, and the change of the network meanwhile influences the future recommendation outcome, we study the co-evolution of the users decision and the recommender system. With this incentive, we make a close study about the users' optimal personalized recommendation algorithm. We consider two benchmark datasets (*Movielens* and *Netflix*) and find that there is a significant enhancement for recommendation performance if all users are assigned with their optimal recommenders. Hence, we design a recommender system that supports multiple recommenders. Moreover, in order to understand the long-term impact of recommender systems, we model the co-evolution of the decisions of users and the recommender system by a rewiring process, and use the structure properties (e.g., Gini coefficient, cluster coefficient) to evaluate long-term diversity of evolution network. Finally, our work highlights that there is a good trade-off between short- and long-term performances of recommendation if online system allows multiple recommenders work at the same time and assigns the optimal recommender to each user in the individual level.

2. Methods and model

2.1. Methods

In this part, we first introduce several well-known recommenders in this work, and then we present the user preference-oriented recommendation method.

Popularity-based recommendation (PR): PR is a classic represent of no-personalized recommendation method, which recommends items to each user based on item popularity. First, it makes statistics on the degree of each item, then sorts the item according to degree in a descend way, and excludes off items that the user have selected. After that, it recommends other products to the user.

User-based collaborative filtering (UCF): The basic idea of UCF is that similar users like similar items. The UCF algorithm follows this process: First, the similarities between the target user and the rest of the users are calculated. Then, the recommendation scores of uncollected items for user i is calculated by

$$P_{i,a} = \sum_{j=1}^{N} s(i,j) a_{j,a} \tag{1}$$

where $s(i, j)$ denotes the similarity between user i and j. Here, we adopt the Jaccard similarity measure. Denote U_i as the neighbor set of user i and k_i as the degree of user i, and the Jaccard index can be expressed as follows:

$$s(i, j) = \frac{\left|U_i \bigcap U_j\right|}{\left|U_i \bigcup U_j\right|}. \tag{2}$$

To obtain a recommendation list for a given user, all items that are currently not connected with this user are sorted according to their recommendation score in a descending order, then the top L items are kept on the recommendation list.

Item-based collaborative filtering (ICF): instead of considering users' similarities, the recommendation score of an item is computed based on the item's similarity with other items collected by a target user. It is defined as follows:

$$P_{i,\alpha} = \sum_{\beta=1}^{N} S(\alpha, \beta) a_{i,\beta}. \tag{3}$$

Like UCF, the similarity equation can be expressed as follows:

$$s(\alpha, \beta) = \frac{\left|U_\alpha \bigcap U_\beta\right|}{\left|U_\alpha \bigcup U_\beta\right|} \tag{4}$$

where U_α is the set of users who have collected item α.

User Preference-oriented Recommendation Method (UPRM): In UPRM, we consider the heterogeneity of users and allow different users use different recommenders in online system simultaneously. The above recommenders (i.e., PR, UCF and ICF) are considered as candidates in UPRM. Due to that different algorithms have different strengths, make them better or worse fit for different users, hence there is no one recommender is one-size-fits-all. The basic idea of UPRM is that recommender system assigns the optimal algorithm to each user in individual level by studying the user's preference, and the optimal algorithm is chosen among candidate algorithms. Specifically, we evaluate the accuracy (e.g., precision) of recommendation to each user by using candidate algorithms, respectively. Then, the optimal algorithm of each user is the one which has the best accuracy among these candidates. Finally, we recommend item to each user by using his/her identified optimal algorithm. The pseudo-code of UPRM algorithm is shown in 1. Please note that our method can extend to other evaluation indexes, such as the Area Under the ROC Curve (AUC) and the Rank Score. Choosing the optimal recommenders by evaluating the AUC and Rank Score indexes is part of our future work.

Algorithm 1 Pseudo-code of UPRM algorithm

Input: The dataset with training-probe sets division
Output: The set of best recommender for each user ($UPRMset$)

for $user_i \in |dataset|$ **do**
  $AccuracyICF(user_i) \leftarrow recommender\_ICF(traindata, probedata, user_i)$
  $AccuracyUCF(user_i) \leftarrow recommender\_UCF(traindata, probedata, user_i)$
  $AccuracyPR(user_i) \leftarrow recommender\_PR(traindata, probedata, user_i)$
  $BestRecommneder(user_i) \leftarrow$
  $min(AccuracyICF(user_i), AccuracyUCF(user_i), AccuracyPR(user_i))$
  Add the $BestRecommneder(user_i)$ to set $UPRMset$
end for
return $UPRMset$;

Matrix factorization approach is not suitable for the unweighted bipartite network, since it is numerical calculation method based on learning algorithm and the predication scores will trend to two extreme values (i.e., 0 and 1) in the unweighted bipartite network.

2.2. Evolution model

To model the co-evolution of user choices and the recommendations generated from the recommender system, we construct a model of recommendation ecosystem as follows. The real data described above are used as the initial configuration. The network evolves through a so-called rewiring process where each link is assigned a time stamp (the initial time stamps are chosen at random). In each rewiring step, the oldest link of every user is redirected to a new item and assigned with the current time (i.e. it becomes the user's newest link). Deleting the oldest links simulates the case where the recommendation results are generated based on recent historical record [25]. When a user follows recommendation, they select an item from their current recommendation list with probability inversely proportional to item rank in the list (the motivation to use rank-reciprocal rather than equal probability for all listed items comes from Ref. [26]). In detail, for user i, the probability of choosing rank-item from recommendation list (RL) is calculated as follows:

$$P(RL_i(\alpha)) = \frac{\frac{1}{RL_i(\alpha)}}{\sum_{\alpha=1}^{L} \frac{1}{RL_i(\alpha)}} \tag{5}$$

where $RL_i(\alpha)$ is the rank of item α in the user i's recommendation list. L is the length of the recommendation list. After that, each user selects an item from the recommendation list according to the probability $P(RL_i(\alpha))$. For example, suppose that the total of macro step is M, the number that user i chooses rank-α item from recommendation list is $M \times P(RL_i(\alpha))$ during the whole process. When a user rejects recommendation, he/she either chooses an item at random (which we refer to as Random Attachment) or real choice of items in dataset. In this paper, we adopt the latter. Please note that he/she chooses an item from his/her uncollected item set in a random way, when a user adopts the Random Attachment method. As the network evolves, degree of each user is preserved. Network structure can be at any moment represented by a so-called network adjacency matrix A, whose element $a_{i\alpha}$ is one when user i is currently connected with item α and zero otherwise.

The computing complexity of the evolution model is dominated by the evolution process. In each evolving iteration, the loop includes four steps: (1) deleting the oldest link according to the time stamp, the computing complexity is $O(|U|)$; (2) running the proposed recommender and generating the recommendation list, the computing complexity depends on what kind of recommender we adopt. For example, the computing complexity of UCF is $O(|U|^2p)$, p is the maximum number of ratings per user,i.e., $p = \max_u |R_u|$; (3) Choosing an item based on the rank reciprocal method, the computing complexity is $O(|U|)$; (4) adding the new link with time stamp into network, the computing complexity is $O(|U|)$. Meanwhile, we define the maximum iteration of evolution model is M, hence the computing complexity of evolution model by using UCF recommender is

$$O(M|U| + |U|^2p + |U| + |U|) \approx O(M|U|^2p). \tag{6}$$

For ICF recommender, the computing complexity of evolution model is

$$O(M|U| + |U|^2p + |U| + |U|) \approx O(M|U|^2p) \tag{7}$$

where q is the maximum number of rating per item, i.e., $q = \max_i |R_i|$.

For PR recommender, the computing complexity of evolution model is

$$O(M(|U| + |U| + |U| + |U|)) \approx O(M|U|). \tag{8}$$

3. Data and metrics

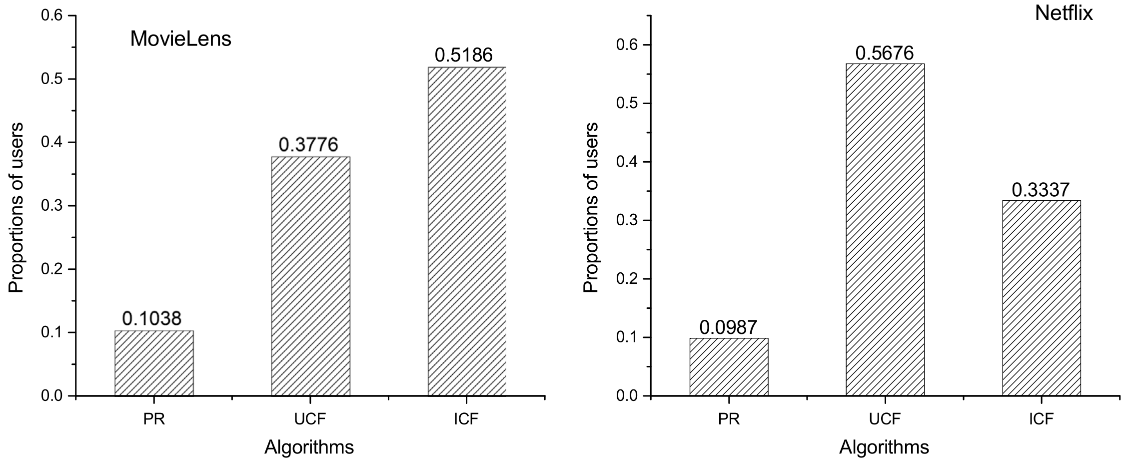
To be our analysis, an online commercial system is described by the user–item bipartite network. Specifically, we use two benchmark datasets: *Movielens* (an online movie rating and recommendation service) and *Netflix* (a DVD rental service). The Movielens data (available at www.grouplens.org) contain 1682 movies and 943 users who have rated movies using the integer scale rating from 1 (worst) to 5 (best). To obtain an unweighted bipartite network, we represent any rating of 3 or more as a link between the respective user and item. The resulting network contains 943 users and 1,574 items with 82,520 links (sparsity is 5.56×10^{-2}). The average degree of users and items is 88 and 52, respectively. The Netflix data is a subset of the original dataset released for the purpose of the Netflix Prize (available at www.netflixprize.com). The subset contains 3000 users and 4692 movies randomly chosen from the original data and all links among them (since the input data uses the same rating scales as *Movielens*, the threshold rating of 3 is also used to determine whether a link is present or not). The resulting network contains 192,966 links and the sparsity is 1.42×10^{-2}. The average degree of user and item is 64 and 41, respectively. To check the recommender, the data is divided into two parts: the training set E_T and the probe set E_P with the ratio of 8:2. The training set is treated as known information, while information in probe set is allowed to be used for prediction.

The recommender can provide each user an ordered list of all his/her uncollected items and may be of interest. A good algorithm is expected to give accurate recommendations, namely to place more items in the probe set in the top of the recommendation list. Here, we first use Precision and Recall to evaluate the accuracy of algorithm, which are two more practical measures.

Precision: For a target user i, the precision of recommendation, $P_i(L)$, is defined as follows:

$$P_i(L) = \frac{hit_i(L)}{L} \tag{9}$$

where $hit_i(L)$ is the number of hitting items (namely the common items exist in the probe set and in the top-L places of recommendation list). Averaging all users' precision, we obtain the mean precision $P(L)$ of the whole system. Here, we set $L = 10$.



(a) MovieLens. (b) Netflix.

Fig. 1. The ratio of optimal algorithm in different dataset.

Recall: the recall of user i, $R_i(L)$, is defined as follows:

$$R_i(L) = \frac{hit_i(L)}{L_i} \tag{10}$$

where L_i is the number of items user i collects in the probe set. As the same as $P(L)$, $R(L)$ is the average of all users' recalls in the whole system.

In order to evaluate the long-term effects of different recommenders, we use the Gini coefficient, clustering coefficient in this paper.

Gini coefficient: it is used to measure inequality of the item popularity distribution during the network evolution. While this quantity has been originally proposed to quantify of income or wealth distribution, it has also been used in other fields [27,28]. The Gini coefficient can be computed as follows:

$$G = \frac{2\sum_{\alpha=1}^{M} \alpha K_\alpha}{M\sum_{\alpha=1}^{M} K_\alpha} - \frac{M+1}{M} \tag{11}$$

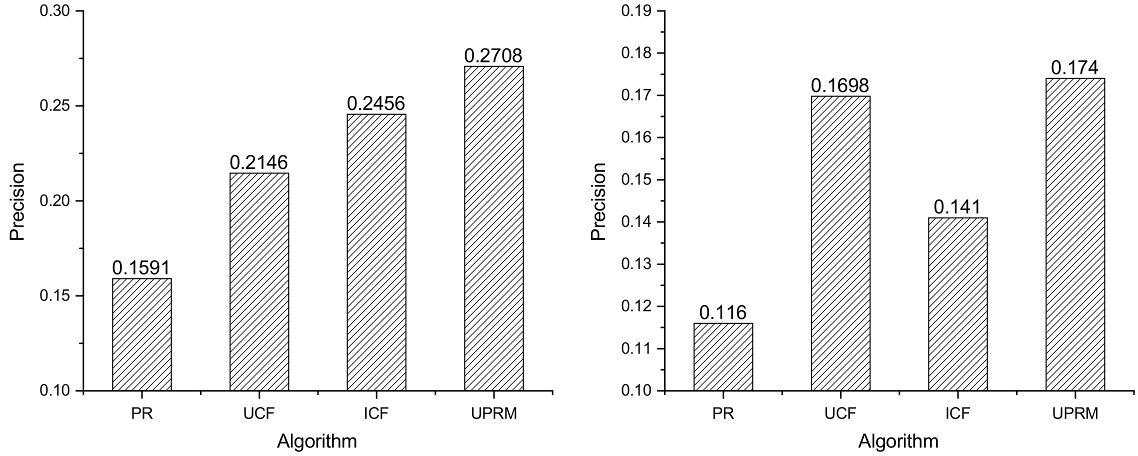
where K_α is the popularity of items, which has been sorted in the ascending order. The two extreme values of the Gini coefficient are 0 and 1 which correspond to equal popularity of all items and zero popularity of all items but one, respectively. An increase of the Gini coefficient thus corresponds to the item popularity distribution becoming more unequal.

Clustering coefficient: it is found that the personalized recommendation mainly relies on the square motifs of the bipartite network [29], hence we adopt the cluster coefficient (CC) to monitor the network evolution. In a bipartite network, the clustering coefficient of node i (CC_i) is calculated by dividing the number of squares (the smallest clique in the bipartite network) passing through i by the number of i's all possible squares.

4. Result

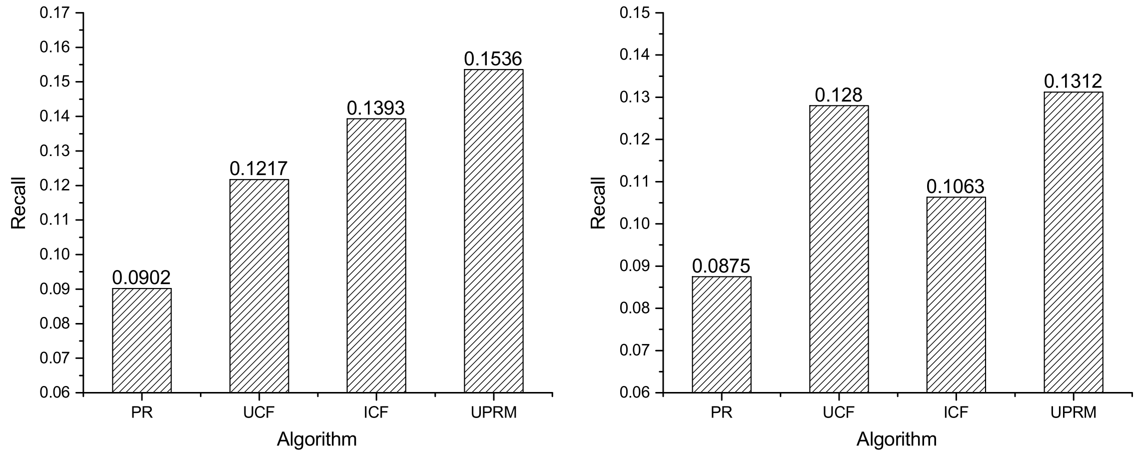
4.1. User heterogeneity and personalized algorithm

In order to design the optimal algorithm for each user, we study the precision of different algorithms works on each user and the statistics results are shown in Fig. 1. It describes the ratio of different algorithms as the optimal recommender in *MovieLens* and *Netflix*, respectively. In *MovieLens*, more than half users (51.86%) chose ICF as his/her recommender. Namely, for these users the ICF has the best precision among candidate algorithms. 37.76% and 10.38% pick the UCF and PR as his/her recommender, respectively. In *Netflix*, UCF is the most favored algorithm (56.76%), followed by ICF (33.37%) and finally the PR (9.87%). We can see that users clearly favor the personalized algorithm over the PR, since PR is a non-personalized and just recommend the most popular items to all users. From Fig. 1, it demonstrates that different algorithms have different strengths, making them a better or worse fit for different users and use cases. Hence, it is necessary to design the personalized recommender system in individual level.



(a) MovieLens. (b) Netflix.

Fig. 2. The precision of each algorithms on different dataset.



(a) MovieLens. (b) Netflix.

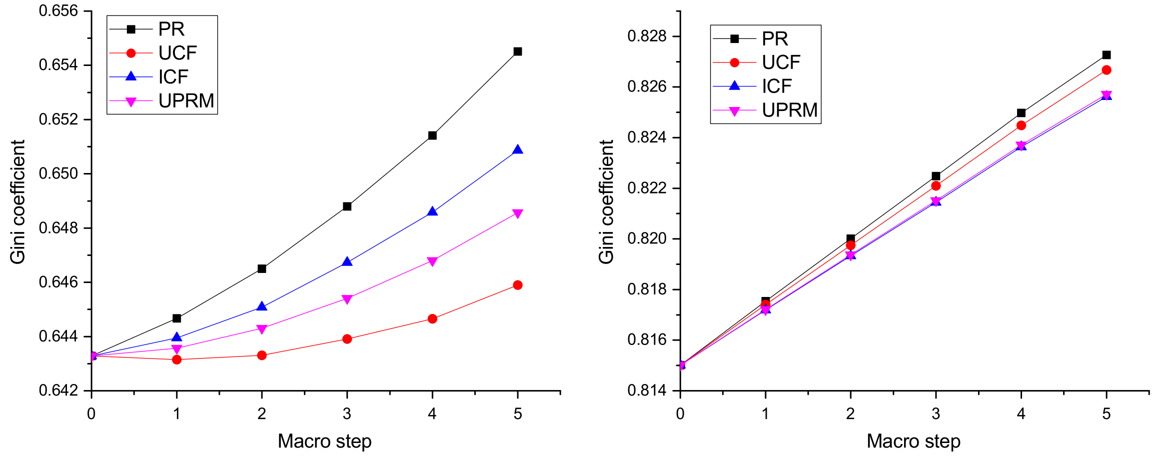
Fig. 3. The recall of each algorithms on different dataset.

4.2. Performance improvement

For testing the accuracy (i.e., precision and recall) of different algorithms and to further remove their possible dependence on the data division, we average precision and recall over five independent training set-probe set divisions. Through the above analysis, the UPRM assigns the best optimal algorithm to each user as his/her recommender. The precision and recall of different algorithms on *MovieLens* and *Netflix* are shown in Figs. 2 and 3 respectively. In *MovieLens*, we can see clearly that the accuracy of UPRM is the best among candidate algorithms either in precision or recall. In detail, compared with PR, UCF and ICF, the precision enhancement of UPRM is 70.29%, 26.19% and 10.26% respectively. The same phenomenon occurs on *Netflix*. Compared with the performance of ICF and UCF, we can find that the accuracy of ICF is better than UCF in *MovieLens*, while the result is just reversed in *Netflix*. However, the accuracy of UPRM is the best either in *MovieLens* or *Netflix*.

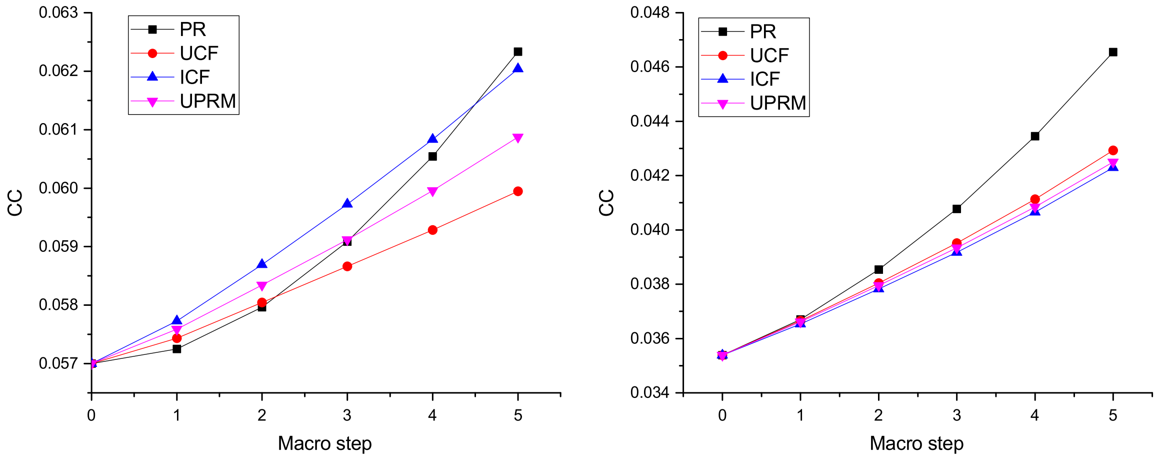
4.3. Long-term impacts

One of the basic topology properties of a complex network is the degree distribution. To obtain a quantitative comparison, we measure the Gini coefficient over the item degrees in each macro-steps of evolution. The higher the value, the more uneven the distribution, as well as the greater the loss of information diversity in the recommender system. The Gini coefficient corresponds to the distributions shown in Fig. 4. The curves of all algorithms increase gradually during the evolution process, which indicate that potential danger of recommender for information diversity, similar to the loss of biodiversity in an online system. Specifically, the PA can lead to a higher Gini coefficient than the others. This is because PR is not personalized and biased on popularity, the recommendation list for different users significantly overlap; as a result,



(a) MovieLens. (b) Netflix.

Fig. 4. Evolution of the Gini coefficient on different dataset.



(a) MovieLens. (b) Netflix.

Fig. 5. Evolution of clustering coefficient (CC) when different recommenders are implemented in real systems.

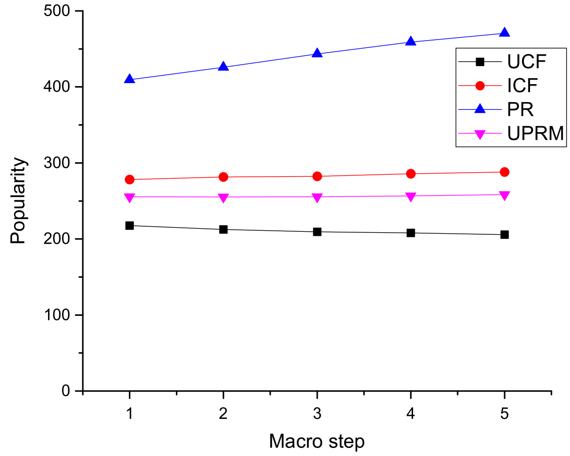
the popularity items will attract a large amount of links. For ICF and UCF, we find that the lower the accuracy is, the lower the Gini is (i.e., the more the diversity is). Namely, it is possibile to compromise recommendation accuracy and long-term impacts on diversity for recommender systems. We also find that although the accuracy of UPRM is the highest among candidates, the Gini of UPRM is not the highest during the evolution processing. It demonstrated that UPRM can get a better trade-off between short-term accuracy and long-term impacts on diversity for recommender systems.

Meanwhile, it is found that the personalized recommender mainly relies on the square motifs of the bipartite networks. Here, we also adopt clustering coefficient (CC) to monitor the network evolution. The value of CC varies from 0 to 1 and a high value indicates a strong local connection. Fig. 5 shows the evolution of CC in different networks where all the recommenders result in a bigger CC than the original real data, since almost all these methods are based on the square motif to realize personal recommender except PR. Among these algorithms, the CC of UPRM is between ICF and UCF, which means that UPRM strengths the local connectivity between different users and trend to recommend novel items at the same time, compared with ICF in *MovieLens* and UCF in *Netflix*.

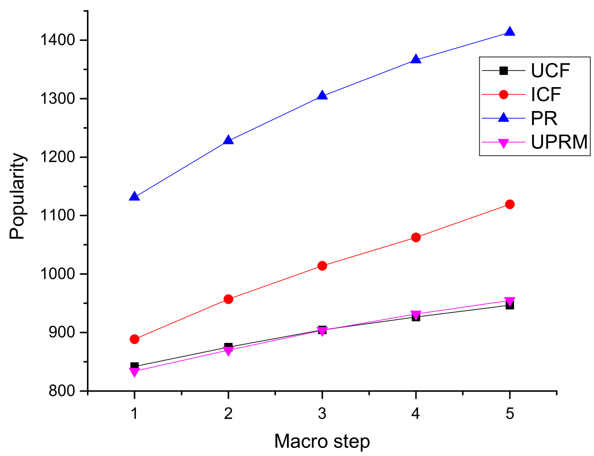
In order to observe the long-term effect of the popularity changes with macro step, we define the popularity by measuring the average item degree $\langle K \rangle$ over all the recommended objects.

$$\langle K \rangle = \frac{1}{nL} \sum_{n}^{u=1} \sum_{\alpha \in O^u} K(\alpha) \tag{12}$$

where o^u is the recommendation list of user u. The lower popularity the system has, the higher novelty the system has. Fig. 6 shows the evolution of popularity of recommendation list produced by different recommenders on each dataset. In Netflix as shown in Fig. 6(b), the curves of all recommenders increase gradually during the evolving process, which indicate



(a) MovieLens.



(b) Netflix.

Fig. 6. Evolution of popularity when different recommenders are implemented in real systems.

that the recommendation novelty of all recommenders decrease as network dynamically evolves. In detail, the popularity of recommendation list produced by PR is the highest among all recommenders, that is because the principle of PR is that recommend items to every user based on uncollected items' popularity (i.e., recommend the most popularity uncollected item to user). The same phenomena also happened in Movielens as shown in Fig. 6(a). For the proposed UPRM recommender, the curve of popularity is between that of ICF and UCF. That means that compared with ICF, UPRM prefer to recommend novel items, while maintaining the recommendation accuracy.

Please note that this paper focuses on the performance of recommenders on the temporal evolving networks. For temporally modeling the co-evolution of user choices and the recommendations generated from the recommender system, we consider the whole original dataset as the initial evolution network, and assign a time stamp to every link. Since we did not divide dataset with time stamp into training-test subset during the evolving process, we cannot evaluate the accuracy of recommender during the evolving process.

5. Conclusion

Recommender system is a very promising technology to address the problem of information overload. Recently, some researcher found that recommender is not one-size-fit-all, different algorithms have different strengths, making them a better or worse fit for different users and use cases. In this paper, we propose a novel personalized recommendation method (UPRM) based on user preference, which applies the recommender to individual level so that each user has his/her own personalized recommender. We find that real users are quite different in their own personalized algorithm, and if each user is assigned with the optimal algorithm, the overall recommendation accuracy can be significantly enhanced. Moreover, in order to study the long-term effects of our proposed algorithm on the evolution of online E-commercial system, we firstly model the mutual feedback between users's decisions and recommender system, and then evaluate the structure indices including Gini coefficient, clustering coefficient and popularity of recommendation list. By conducting experiments on the real dataset, we find that the recommenders have reinforcing effects on these indices. Specifically, the Gini coefficient, clustering coefficient and popularity of recommendation list are increased with model evolving in the long term. However, we remark that the UPRM can get a good trade-off between the short-term recommendation accuracy and long-term diversity.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61370150, 61433014 and 61602434), the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2015jcyjB0244 and the Sichuan Provincial Science and Technology Department (2015GZ0143).

References

[1] J.B. Schafer, J.A. Konstan, J. Riedl, E-commerce recommendation applications, Appl. Data Min. Electron. Commer. (2001) 115–153.
[2] C.N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in: Proceedings of the 14th International Conference on World Wide, 2005, pp. 22–32.
[3] D. Goldberg, D. Nichols, B.M. Oki, D. Terry, Using collaborative filtering to weave an information tapestry, Commun. ACM 35 (12) (1992) 61–70.
[4] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The Adaptive Web, 2007, pp. 291–324.

[5] Y-D. Zaho, S-M. Cai, M. Tang, M-S. Shang, A fast recommendation algorithm for social tagging systems: a delicious case, 2015. arXiv preprint arXiv:1512.08325.
[6] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, 2007, pp. 325–341.
[7] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: A constant time collaborative filtering algorithm, Inf. Retr. 4 (2) (2001) 133–151.
[8] S. Maslov, Y.C. Zhang, Extracting hidden information from knowledge networks, Phys. Rev. Lett. 87 (24) (2001) 248701.
[9] T. Hofmannn, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 89–115.
[10] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[11] P. Laureti, L. Moret, Y.C. Zhang, Y.K. Yu, Information filtering via iterative refinement, Europhys. Lett. 75 (6) (2006) 1006.
[12] C. De Kerchove, P. Van Dooren, The PageTrust algorithm: How to rank web pages when negative links are allowed? in: SDM, 2008, pp. 346–352.
[13] G. Cimini, D. Chen, M. Medo, L. Lü, Y.C. Zhang, T. Zhou, Enhancing topology adaptation in information-sharing social networks, Phys. Rev. E 85 (4) (2012) 046108.
[14] M.D. Ekstrand, D. Kluver, F.M. Harper, J.A. Konstan, Letting users choose recommender algorithms: An experimental study, in: Proceedings of the 9th ACM Conference on Recommender Systems, 2015, pp. 11–18.
[15] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. 107 (10) (2010) 4511–4515.
[16] Y. Guan, D. Zhao, A. Zeng, M.S. Shang, Preference of online users and personalized recommendations, Physica A 392 (16) (2013) 3417–3423.
[17] J.G. Liu, T. Zhou, H.A. Che, B.H. Wang, Y.C. Zhang, Effects of high-order correlations on personalized recommendations for bipartite networks, Physica A 389 (4) (2010) 881–886.
[18] X. Zhu, H. Tian, S-M Cai: Personalized recommendation with corrected similarity, J. Stat. Mech. Theory Exp. 7 (2014) P07004.
[19] Q. Guo, W.J. Song, J.G. Liu, Ultra-accurate collaborative information filtering via directed user similarity, Europhys. Lett. 107 (1) (2014) 18001.
[20] J.G. Liu, K. Shi, Q. Guo, Solving the accuracy-diversity dilemma via directed random walks, Phys. Rev. E 85 (1) (2012) 016118.
[21] J.G. Liu, T. Zhou, B.H. Wang, Y.C. Zhang, Q. Guo, Effects of user's tastes on personalized recommendation, Internat. J. Modern Phys. C 20 (12) (2009) 1925–1932.
[22] M.S. Shang, L.L, Y.C. Zhang, T. Zhou, Empirical analysis of web-based user-object bipartite networks, Europhys. Lett. 90 (4) (2010) 48006.
[23] A. Zeng, C.H. Yeung, M. Medo, Y.C. Zhang, Modeling mutual feedback between users and recommender systems, J. Stat. Mech. Theory Exp. 7 (2015) 07020.
[24] D.D. Zhao, A. Zeng, M.S. Shang, J. Gao, Long-term effects of recommendation on the evolution of online systems, Chin. Phys. Lett. 30 (11) (2013) 118901.
[25] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (2010) 89–97.
[26] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, G. Gay, Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search, AMC Trans. Inf. Syst, 25 (2) (2007) 7.
[27] L. Wittebolle, M. Marzorati, L. Clement, A. Balloi, D. Daffonchio, K. Heylen, P. De Vos, W. Verstraete, N. Boon, Initial community evenness favours functionality under selective stress, Nature 458 (7238) (2009) 623–626.
[28] W. Halffman, L. Leydesdorff, Is inequality among universities increasing? Gini coefficients and the elusive rise of elite universities, Minerva 48 (1) (2010) 55–72.
[29] Z. Huang, D.D. Zeng, H. Chen, Analyzing consumer-product graphs: Empirical findings and applications in recommender systems, Manag. Sci. 53 (7) (2007) 1146–1164.