

Preference of online users and personalized recommendations

Yuan Guan^a, Dandan Zhao^a, An Zeng^{b1}, Ming-Sheng Shang^{a,c2}

^aWeb Science Center, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China

^bDepartment of Physics, University of Fribourg, Chemin du Musée 3, CH-1700 Fribourg, Switzerland

^cInstitute of Information Economy, Alibaba Business School, Hangzhou Normal University, Hangzhou, 310036, PR China

Abstract

In a recent work [Proc. Natl. Acad. Sci. 107 (2010) 4511], a personalized recommendation algorithm with high performance in both accuracy and diversity is proposed. This method is based on the hybridization of two single algorithms called probability spreading and heat conduction, which respectively are inclined to recommend popular and unpopular products. With a tunable parameter, an optimal balance between these two algorithms in system level is obtained. In this paper, we apply this hybrid method in individual level, namely each user has his/her own *personalized hybrid parameter* to adjust. Interestingly, we find that users are quite different in personalized hybrid parameters and the recommendation performance can be significantly improved if each user is assigned with his/her optimal personalized hybrid parameter. Furthermore, we find that users' personalized parameters are negatively correlated with users' degree but positively correlated with the average degree of the items collected by each user. With these understandings, we propose a strategy to assign users with suitable personalized parameters, which leads to a further improvement of the original hybrid method. Finally, our work highlights the importance of considering the heterogeneity of users in recommendation.

Keywords: Hybrid algorithm, Recommender system, Personalized parameter, User heterogeneity

1. Introduction

The last few years have witnessed serious problems of information explosion. The amount of information, especially on internet, is increasing far more quickly than our ability to process it. For instance, there are thousands of movies, millions of books, billions of web pages. The ongoing rapid expansion of the Internet resource greatly increases the necessity of effective tools for filtering the abundant information [1]. In this context, recommender systems rise. They uncover users' potential interests according to their historical activities rather the specified keywords provided by users. So far, numerous recommendation algorithms based on different ideas and concepts have been proposed, including collaborative filtering [2, 3], the content-based method [4], spectral analysis [5, 6], latent semantic models and Dirichlet allocation [7, 8], iterative self-consistent refinement [9, 10, 11] as well as topology adaptation method [12].

Recently, some diffusion-based recommendation algorithms, such as the heat conduction (HeatS) [13, 14] and probability spreading (ProbS) [15], have been applied in personalized recommendation. The ProbS method is inclined to recommend popular products while the HeatS method tends to recommend cold products for the individual target user. Soon after that, the hybrid method (HHP) combining heat conduction and probability spreading algorithm is proposed to achieve better recommendation performances [16]. 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. Also, the performance of diffusion-based recommendation methods have been enhanced by the recently proposed preferential diffusion process [17], biased heat conduction [18] and network manipulation [19].

¹an.zeng@unifr.ch

²Shang.mingsheng@gmail.com

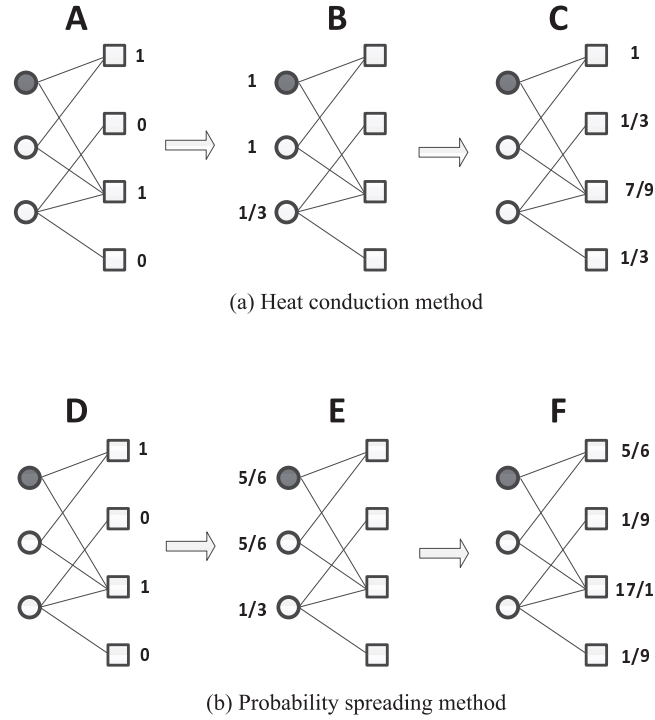


Figure 1: The illustration of the (a) HeatS and (b) ProbS algorithms on the bipartite user-object network. Users are shown as circles; objects are squares. The target user is indicated by the shaded circle.

In this paper, we put forward a recommendation algorithm which makes further improvement to HHP. In the original HHP, the optimal hybrid parameter takes a universal value, under which the system achieves the optimal balance of recommendation for the cold items and the popular items. That's to say, all users in the system have to adopt the same hybrid parameter. However, HHP method doesn't consider the heterogeneity of real users. Actually, the hybrid algorithm can be applied in individual level where each user can have his/her own *personalized hybrid parameter* to adjust. Specifically, if a user likes to collect the popular items, the hybrid method should give more weight to the ProbS algorithm, since ProbS is good at recommending popular items, and vice versa.

With this incentive, we make a close study in this paper about the users' optimal personalized hybrid parameters. We consider two benchmark data sets (*MovieLens* and *Delicious*) and find that there is a significant enhancement for recommendation performance if all users are assigned with their optimal personalized hybrid parameters. We also find that users have quite different optimal personalized hybrid parameters. Moreover, we show that most users' personalized hybrid parameters are negatively correlated with their degrees but positively correlated with the average degrees of items collected by them (denoted as d). Finally, we propose a user-oriented HHP algorithm (UHHP) in which each user's personalized hybrid parameter is assigned according to his/her d . The simulation shows that UHHP can further improve the performance of the original HHP algorithm. Finally, our work highlights the importance of considering the heterogeneity of users in recommendation.

2. Data and Metrics

We use two benchmark data sets to carry out our analysis, *MovieLens* and *Delicious*. The *MovieLens* data contains real data of 100,000 ratings from 943 users on 1,682 movies. Then a coarse-graining process is used to transform these ratings into unary form: an item is considered to be collected by a user only if the given rating is three or more, where five is the maximum rating. After the coarse gaining process, there remains 943 users and 1,574 items with 85,250 edges (sparsity is $5.20 \cdot 10^{-2}$). The *Delicious* data is inherently unary to indicate whether a user has either

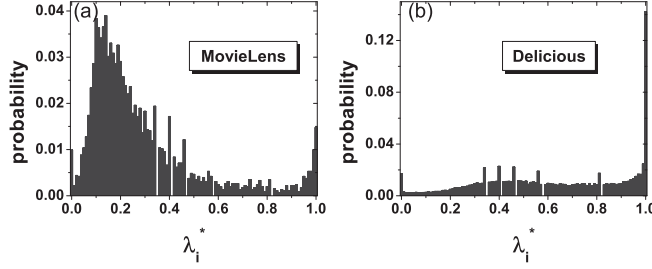


Figure 2: the distribution of λ_i^* in (a) *MovieLens* and (b) *Delicious* data sets.

collected a web link or not. It contains 9,998 users and 232,657 items with 1,233,995 edges (sparsity is $5.30 \cdot 10^{-4}$). To check the recommendation algorithm, the data is divided into two parts: the training set E_T and the probe set E_P with the ratio of 9 : 1. The training set is treated as known information, while no information in probe set is allowed to be used for prediction.

The recommendation algorithms can provide each user an ordered list of all his/her uncollected items. 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 Rank Score [15] to measure the ability of a recommendation algorithm to produce a good ordering of items that matches the user's preference. Since real users usually consider only the top part of the recommendation list, two more practical measures are also employed, namely Precision and Recall [20].

Rank Score: Recommendation algorithms can provide each active user with an ordered list of his/her uncollected items. For a target user i , we calculate the position for each of his/her link in the probe set. For example, if there are 1000 uncollected items for i and α is the 30th from the top in the ordered list, we say the position of α is the top 30/1000, and thus the rank score $RS_{i\alpha} = 0.03$. Averaging the rank score over all the pairs in the probe set, we obtain the final rank score RS for the whole system. A good algorithm is expected to have a smaller RS . Besides, we consider the rank score of cold items (i.e. items with small degrees). Since their information is little, it is normally difficult to accurately recommend them. This is often referred as the well-known *cold start* problem. We use $RS_{k \leq x}$ to denote the average rank score of the items with degree smaller than x . In our work, x is set to be 10 according to previous researches [21].

Precision: For a target user i , the precision of recommendation, $P_i(L)$, is defined as $P_i(L) = \frac{h_i(L)}{L}$, where $h_i(L)$ indicates 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' precisions, we obtain the mean precision $P(L)$ of the whole system. In this paper, we set $L = 20$.

Recall: In the same way, the recall of user i , $R_i(L)$, is defined as $R_i(L) = \frac{h_i(L)}{L_i}$, where $h_i(L)$ indicates the number of hitting items and L_i is the number of items user i collects in the probe set. Averaging all users' recalls, we obtain the mean recall $R(L)$ of the whole system. Again, we set $L = 20$.

3. User heterogeneity and personalized parameter

Let's consider an online system with $|U|$ users and $|O|$ items. It can be represented by a bipartite network with adjacency matrix A , where the element $A_{i\alpha} = 1$ if user i has collected item α , and $A_{i\alpha} = 0$ otherwise. Then we can calculate the degree of user and item as $k_i = \sum_{\alpha \in O} A_{i\alpha}$ and $k_\alpha = \sum_{i \in U} A_{i\alpha}$, respectively. (Throughout this paper we use Greek and Latin letters, respectively, for item- and user-related indices).

First, we briefly review the original hybrid algorithm HHP proposed in [16]. The HHP algorithm works by assigning items with initial amount of "resources" denoted by the vector \mathbf{f} , and then redistributing them via the transformation $\tilde{\mathbf{f}} = W\mathbf{f}$, where W is the transition matrix containing the mechanism of resource reallocation. According to the resources finally got by items, a list of sorted items is generated for users. Therefore, the definition of the transition matrix, W , is a vital factor. In order to take the advantages of heat conduction [13] and probability spreading [15], the

HHP method incorporates a hybrid parameter λ into the transition matrix as

$$W_{\alpha\beta} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^\lambda} \sum_{j \in U} \frac{A_{\alpha j} A_{\beta j}}{k_j}, \quad (1)$$

where $\lambda = 0$ corresponds to the pure HeatS algorithm, while $\lambda = 1$ is the pure ProbS algorithm. Fig. 1 gives an example of the resource spreading processes with ProbS and HeatS algorithms on a user-object bipartite network. By tuning λ to a certain value, the HHP method can achieve its best performance.

Actually, it has already been shown that popular items have generally higher recommendation scores in ProbS algorithm while the HeatS algorithm usually gives cold items higher recommendation scores [22]. With the hybrid parameter, the HHP method is adjusting whether to recommend more popular items or cold items. In HHP method, each user has the same hybrid parameter, i.e., all the users are assumed to have the same tendency to ProbS and HeatS. However, this assumption is not true in real case since some users may prefer hot items while some others may like the niche items more. Therefore, we apply the hybrid method to the individual level, i.e., each user can adjust his/her own *personalized hybrid parameter* λ_i for $W_{\alpha\beta}$ to obtain the best recommendation for him/her. The optimal λ_i is obtained when the rank score of the user i is minimized. It's worth noting that there are few users whose rank score change with λ_i in a zigzag way. In this case, we average all the λ_i corresponding to the lowest rank score to determine the λ_i^* . In table 1, we report the system results of rank score, precision, and recall if each user is assigned with the optimal personalized hybrid parameter λ_i^* (which we refer as the Ideal_HHP case). The results in table 1 indicate that the recommendation accuracy can be significantly improved.

Table 1: The improvement of Ideal_HHP compared to the original HHP algorithm in terms of three metrics including $RS_{k \leq 10}$, Rank Score, Precision, and Recall.

	Algorithm	$RS_{k \leq 10}$	RS	$P(20)$	$R(20)$
<i>MovieLens</i>	HHP	0.4545	0.0901	0.1281	0.3132
	IdealHHP	0.3853	0.0836	0.1370	0.3809
	Enhancement	15.23%	7.21%	6.95%	21.62%
<i>Delicious</i>	HHP	0.3239	0.1998	0.0096	0.0162
	IdealHHP	0.3007	0.1944	0.0107	0.0327
	Enhancement	7.16%	2.7%	11.46%	101.85%

We further move to investigate the property of user's optimal personalized hybrid parameter λ_i^* . In fig. 2, we show the distribution of λ_i^* in *MovieLens* and *Delicious*. In *MovieLens*, there are two peaks. The bigger one is close to $\lambda_i^* = 0.1$ while the smaller one is near $\lambda_i^* = 1$. In *Delicious*, there is an obvious peak in $\lambda_i^* = 1$ and other λ_i^* are almost uniformly distributed between 0 and 1. These results indicate that users have quite different personalized hybrid parameters in real systems. If we use the same hybrid parameter for all users as the original HHP, many users cannot receive the best recommendations.

In order to design a method to predict users' optimal personalized hybrid parameters, we study the correlation between λ_i^* and two measures of users' behaviors. The first one is simply the degree of a user (k_i), and the other one is the average degree of the items collected by a user (d_i). In fig. 3(a),(b) and fig. 4(a),(b), we can see that λ_i^* is positively correlated with d_i and negatively correlated with k_i . From the correlation relationship mentioned above, we can get some inspirations. For the users with smaller d , they are generally more interested in niche items. A smaller λ_i^* , which means giving more weight to HeatS algorithm, works better in this case. For the users with larger d , the hybrid method should give more weight to the ProbS algorithm, so that a larger λ_i^* is used. In ref. [23, 24], it is found that users with large degree usually have small d , and vice versa. Users with large degree have already collected almost all popular items and they are more likely to pick up niche items in the future. Therefore, it is better to assign active users with relatively small personalized hybrid parameters so that more cold items will be included in the recommendation list. For the users with smaller degree, they are mainly trend followers and a larger personalized hybrid parameters will lead to more popular items in the recommendation list.

We also consider the distributions of d and k . The results are reported in fig. 3(c), (d) and fig. 4(c), (d). As expected, the user degree distribution (distribution of k) is heterogeneous while d is found to be more homogeneously

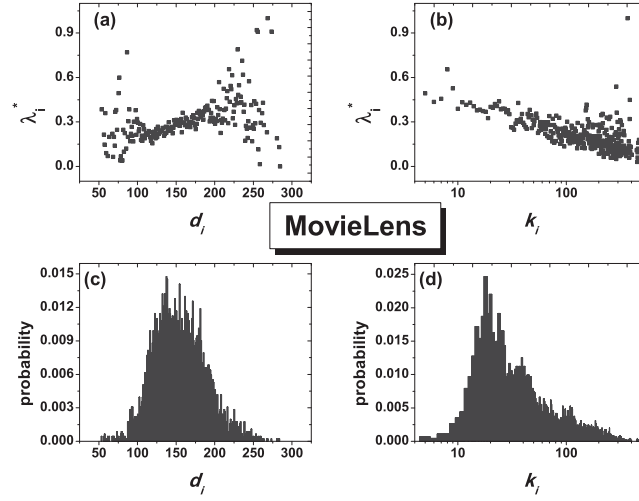


Figure 3: (a) The correlation between λ_i^* and d_i (with Pearson correlation coefficient of 0.4693). (b) The correlation between λ_i^* and k_i (with Pearson correlation coefficient of -0.5424). (c) The distribution of k_i . (d) The distribution of d_i . The data in this figure is *MovieLens*.

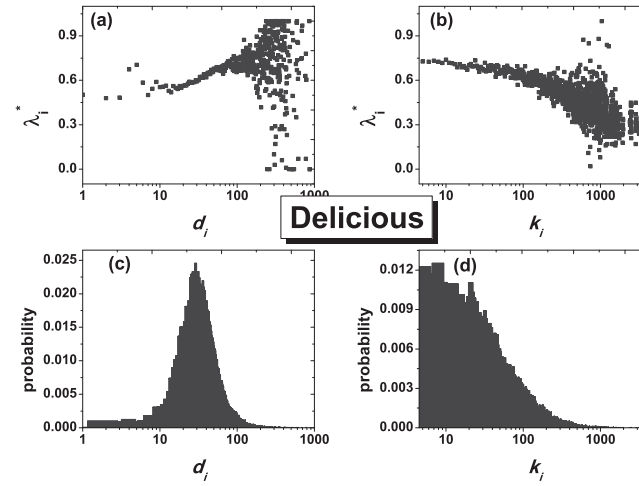


Figure 4: (a) The correlation between λ_i^* and d_i (with Pearson correlation coefficient of -0.0271). (b) The correlation between λ_i^* and k_i (with Pearson correlation coefficient of -0.6395). (c) The distribution of k_i . (d) The distribution of d_i . The data in this figure is *Delicious*.

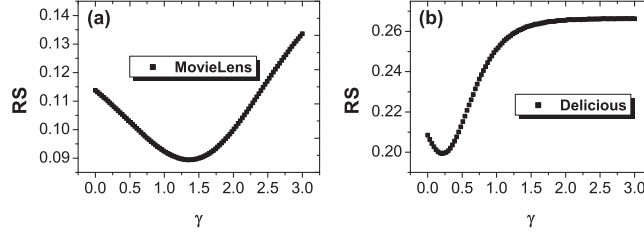


Figure 5: Performance of the proposed hybrid algorithm UHHP on the two different data sets in terms of RS .

distributed. We recall that the distribution of λ_i^* is closer to a homogeneous distribution as shown in fig. 2. Therefore, in the next section, we will propose a strategy to assign each user a suitable personalized hybrid parameter λ_i^* based on user's d value.

4. Recommendation with personalized algorithm

According to the analysis above, we propose a user-based hybrid algorithm (UHHP) as

$$W_{\alpha\beta} = \frac{1}{k_{\alpha}^{1-\lambda_i} k_{\beta}^{\lambda_i}} \sum_{j \in U} \frac{A_{\alpha j} A_{\beta j}}{k_j}, \lambda_i = \left(\frac{d_i}{d_{max}}\right)^{\gamma}, \quad (2)$$

where λ_i is the users' personalized hybrid parameter, d_i is the average item degree collected by the user, d_{max} is the max degree of the items, and γ is a tunable parameter. By varying the value of γ , we can adjust the degree of correlation between λ_i and d_i . When $\gamma=0$, λ_i equals to 1 for all users in the system. And when γ increases, λ_i gradually becomes heterogeneous according to d_i .

Like HHP, the proposed UHHP algorithm is also based on the three-step diffusion process. In the original hybrid method, the parameter λ adjusts the amount of resources finally obtained by different items. When $\lambda = 1$, large degree items incline to receive more resources. On the contrary, small degree items will receive more resources when $\lambda = 0$. In order to do recommendation for all N users, the three-step diffusion needs to originate from each individual user. In HHP, the same λ is applied for all users. It means that the final resources for items are adjusted based on the same λ in all these N times of diffusion. Obviously, it is inevitable that the resource distribution over items is not tuned to the optimum in some diffusion. In the UHHP method, each user has his own λ_i . This modification largely increases the freedom for parameters adjustment. Therefore, the UHHP may outperform the original hybrid method.

The Rank Score metric is independent of recommendation list length, and is one of the most important indicators to evaluate a recommender system. In fig. 5, we show the dependence of Rank Score on γ . As we can see, there is an optimal γ under which the UHHP can achieve its best performance in Rank Score. We can obtain the optimal parameter, $\gamma^* = 1.36$ and $\gamma^* = 0.21$, for the *MovieLens* and the *Delicious* data sets, respectively. In the optimal case, we compare each user's personalized hybrid parameter of UHHP and that of the Ideal_HHP, which can be seen from fig. 6. We conduct that the personalized hybrid parameter generated from UHHP are very much close to that of Ideal_HHP. In the following, we will compare the recommendation performance of UHHP method under these optimal γ^* to the corresponding optimal case in HHP. Apart from Rank Score, we also consider two more realistic metrics which depend on recommendation list length: Precision and Recall.

The results for the UHHP algorithm are given in Table 2. We find the UHHP algorithm performs better in all metrics mentioned. Take *MovieLens* for example, with comparative performance on Rank Score RS , metrics including $RS_{k \leq 10}$, Precision $P(20)$ and Recall $R(20)$ in the UHHP method can have an enhancement of 4.29%, 1.56% and 4.76%. As for *Delicious* data, UHHP method only slightly outperforms in terms of RS and $RS_{k \leq 10}$ than HHP. However, it can have significant improvement in aspects of $P(20)$ and $R(20)$ with an enhancement of 8.33% and 36.42%, respectively. Actually, improving Precision and Recall is more significant for real users' satisfactions, since real users usually consider only the top part of the recommendation list, while ignore most items appearing in the behind.

In the above, we divided the data into training set and probe set with the ratio of 9 : 1. We actually examined the effects of data sparsity on the algorithmic performance of the four metrics mentioned above. The results show that

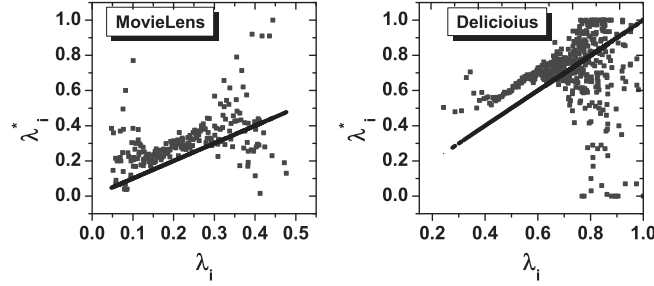


Figure 6: The correlation of the user's personalized hybrid parameter between UHHP and Ideal_HHP. The diagonal lines are plotted to guide eyes.

Table 2: Comparison of HHP with UHHP algorithm, in terms of $RS_{k \leq 10}$, Rank Score, Precision, and Recall.

	Algorithm	Parameters	$RS_{k \leq 10}$	RS	$P(20)$	$R(20)$
<i>MovieLens</i>	HHP	$\lambda = 0.17$	0.4545	0.0901	0.1281	0.3132
	UHHP	$\gamma = 1.36$	0.4350	0.0894	0.1301	0.3281
	Enhancement		4.29%	0.78%	1.56%	4.76%
<i>Delicious</i>	HHP	$\lambda = 0.51$	0.3239	0.1998	0.0096	0.0162
	UHHP	$\gamma = 0.21$	0.3228	0.1993	0.0104	0.0221
	Enhancement		0.34%	0.25%	8.33%	36.42%

the advantage of the UHHP methods decreases as the training set gets small. This is reasonable because the UHHP method needs enough data to accurately estimate each user's preference.

5. Conclusion

Recommender system is a very promising technology to address the problem of information abundance. Recently, a hybrid method (HHP) combining the Heat conduction (HeatS) and Probability spreading (ProbS) algorithm was proposed. The ProbS algorithm mainly recommends popular items and thus has accurate recommendation. On the contrary, the HeatS algorithm tends to recommend niche items and enjoys quite diverse recommendation. With a tunable parameter, the hybrid method can achieve a high performance in both recommendation accuracy and diversity. In this paper, we apply the original hybrid method to individual level so that each user has a personalized hybrid parameter to adjust. We find that if each user is assigned with the optimal personalized hybrid parameter, the overall recommendation accuracy can be significantly enhanced. Moreover, we find that real users are quite different in their optimal personalized hybrid parameters. Besides, the personalized hybrid parameter is found to be negatively correlated with user's degree but positively correlated with the average item degree collected by the user.

Additionally, we propose a user-based hybrid method (UHHP). In UHHP, each user is assigned with a personalized hybrid parameter according to the average degree of items selected by him/her. We test the UHHP method in two benchmark data sets (*MovieLens* and *Delicious*) and find that our method can further improve the recommendation accuracy (especially the Precision and Recall) compared with the HHP algorithm. However, how to accurately estimate the optimal tunable parameters in recommender systems for future recommendation is still a challenge. So far, the usual way to solve the problem is based on the analysis of the historical data. Normally, the historical data is divided into training set and probe set. The parameter for future recommendation is determined when the algorithm achieves its best performance in this training-probe set division. In our work, we also adopted this way to estimate our optimal gamma.

Besides, in real systems, the correlation between λ_i^* and d_i may significantly change from one network to another. That is why the optimal gamma is different in different networks. In our simulation, we find that $\gamma = 1.36$ and $\gamma = 0.21$ are the optimal values for *MovieLens* and *Delicious* respectively. It means that the power of the correlation between λ_i^* and d_i in *MovieLens* is higher than that in *Delicious*. However, we did not find a clear correspondence of

the network structure to the optimal gamma so far. This would be an interesting topic for our future study. Moreover, in our experiments, we do not take time information into consideration. When the data is provided along with the time information, it's better to use the recent average item degree to predict his optimal λ_i^* .

Finally, we remark that the current method to estimate the user's personalized hybrid parameter is not the best one. For example, analyzing users' historical activity records with time information may lead to a deeper understanding of users' behavior pattern [25] and thus a better prediction of their personalized hybrid parameters. Moreover, users' optimal personalized hybrid parameters will change with time in real systems. For some period, a user might like popular products. After some time, he/she might shift to some relevant but not so popular products. In this situation, the method should also take into account the temporal behavior pattern of users. These problems ask for further investigations in the future.

Acknowledgement

We thank Tao Zhou for the useful suggestions. This work is supported by the National Science Foundation of China under Grant No.60973069, No. 11105024 and No. 61073099.

References

- [1] L. Lü, C. H. Yueng, M. Medo, Y.-C. Zhang, Z.-K. Zhang and T. Zhou, Phys. Rep. 519 (2012) 1.
- [2] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, Commun. ACM 35 (1992) 61.
- [3] J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen, Lect. Notes Comput. Sc. 4321 (2007) 291.
- [4] M. J. Pazzani and D. Billsus, Lect. Notes Comput. Sc. 4321 (2007) 325.
- [5] K. Goldberg, T. Roeder, D. Gupta and C. Perkins, Inform. Retrieval 4 (2001) 133.
- [6] S. Maslov and Y.-C. Zhang, Phys. Rev. Lett. 87 (2001) 248701.
- [7] T. Hofmann, ACM T. Inform. Syst. 22 (2004) 89.
- [8] D. M. Blei, A. Y. Ng and M. I. Jordan, J. Mach. Learn. Res. 3 (2003) 993.
- [9] P. Laureti, L. Moret, Y.-C. Zhang and Y.-K. Yu, Europhys. Lett. 75 (2006) 1006.
- [10] C. de Kerchove and P. Van Dooren, SIAM News 41 (2008) 1.
- [11] J. Ren, T. Zhou and Y.-C. Zhang, Europhys. Lett. 82 (2008) 58007.
- [12] G. Cimini, D.-B. Chen, M. Medo, L. Lü, Y.-C. Zhang and T. Zhou, Phys. Rev. E 85 (2012) 046108.
- [13] Y.-C. Zhang, M. Blattner and Y.-K. Yu, Phys. Rev. Lett. 99 (2007) 154301.
- [14] A. Stojmirovic and Y.-K. Yu, J. Comput. Biol. 14 (2007) 1115.
- [15] T. Zhou, J. Ren, M. Medo and Y.-C. Zhang, Phys. Rev. E 76 (2007) 046115.
- [16] T. Zhou, Z. Kuscik, J.-G. Liu, M. Medo, J.R. Wakeling and Y.-C. Zhang, Proc. Natl. Acad. Sci. 107 (2010) 4511.
- [17] L. Lü, W. Liu, Phys. Rev. E 83 (2011) 066119.
- [18] J.-G. Liu, T. Zhou, and Q. Guo, Phys. Rev. E 84 (2011) 037101.
- [19] F.-G. Zhang and A. Zeng, Europhys. Lett. 100 (2012) 58005.
- [20] J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl, ACM Trans. Inf. Syst. 22 (2004) 5.
- [21] T. Qiu, G. Chen, Z.-K. Zhang and T. Zhou, Europhys. Lett. 95 (2011) 58003.
- [22] A. Zeng, C. H. Yueng, M.-S. Shang and Y.-C. Zhang, Europhys. Lett. 97 (2012) 18005.
- [23] M.-S. Shang, L. Lü, Y.-C. Zhang and T. Zhou, Europhys. Lett. 90 (2010) 48006.
- [24] C.-J. Zhang and A. Zeng, Physica A 391 (2012) 1822.
- [25] C. Song, T. Koren, P. Wang, A.L. Barabasi, Nature Phys. 6 (2010) 818.