

MEASURING ON-LINE USERS PREFERENCE AND PERSONALIZED RECOMMENDATIONS*

HAO LIAO^{a,b}, JING SHEN^{a,b}, RUI XIAO^b, FEI YU^b, MINHUA LU^c

^aGuangdong Province Key Laboratory of Popular High Performance Computers
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen 518060, China

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

^cNational-Regional Key Technology Engineering Laboratory for Medical
Ultrasound, Guangdong Key Laboratory for Biomedical Measurements and
Ultrasound Imaging, Department of Biomedical Engineering, Shenzhen University
Shenzhen 518060, China

(Received February 29, 2016)

Improving the personalization of recommendation methods is a hot topic with wide application in real on-line commercial systems. One major concern is that an algorithm that focuses too strongly on diversity is putting recommendation accuracy at risk. Based on the method described in [*Proc. Natl. Acad. Sci. USA* **107**, 4511 (2010)], we propose a more personalized algorithm in which each user is assigned with a parameter for the initial configuration setting and a parameter for the hybridization. We find that each user has his/her optimal parameters which are very different from user to user. We finally design a simple method to estimate users' personalized parameters and the recommendation accuracy can be improved accordingly.

DOI:10.5506/APhysPolBSupp.9.109

1. Introduction

The digital revolution brought to us what is known as “information overload”: there is too much information for a single individual to deal with. As a result, nowadays there is hardly an e-commerce website without some form of information filtering and recommendation service [1]. Thanks to the Web 2.0 and Web applications, the recommender systems have been achieving rapid development. The recommender systems can help the user to find

* Presented at the Summer Solstice 2015 International Conference on Discrete Models of Complex Systems, Toronto, Ontario, Canada, June 17–19, 2015.

the useful items from the information ocean. The e-commerce development has also greatly promoted the advantages of recommender systems, such as the `Amazon.com` and `eBay.com`. Accurate and efficient recommendation algorithm can help us analyze the potential consumption trends of users, and eventually provide an effective personalized service for them. Collaborative filtering (CF) is the most popular applied technology in the recommender systems [2–4]. However, the classical CF algorithm only takes into account the effect of similar users or items which will lead to the recommendation results become more and more similar for each user. In addition, the content-based [5], trust-aware [6], social impact [7, 8] and tag-aware [9] are also frequently used recommender technologies.

Recently, the fruitful achievements of complexity theory, especially some physical methods such as mass diffusion [10] and heat conduction [11], have attracted an increasing attention from both computer science and physical community. Researchers use the bipartite network to recommend and solve various fundamental questions in both research and application [12–16]. In fact, the mass diffusion algorithm is a random walk process, which has high accuracy but low personality and diversity, the heat conduction method has low accuracy but high personality and diversity. In Ref. [17], the authors proposed a hybrid method to combine the mass diffusion and heat conduction which solve the apparent diversity-accuracy dilemma of recommender systems. In other words, the recommender systems will not only consider to recommend the popular objects, but also the niche objects, which indicates the personality and diversity play an important role in evaluating recommender systems.

After [17], many different methods are proposed to achieve even better recommendation performance. For example, the preferential diffusion [18] and the biased heat conduction [19] has been designed to yield a higher accuracy and larger diversity compare to the method in [20]. Moreover, the network manipulation has been shown to effectively solve the cold-start problem in recommendation [21, 22]. To enhance the efficiency of the recommendation process, the method to extract the information backbone (minimum structure) from on-line system is also designed [23]. Very recently, the long-term influence of the recommendation methods on the user-item bipartite network evolution is studied [24]. It is found that many personalized recommendation methods have reinforced the effect on item degree distribution.

Generally speaking, personalization is crucial to generate diverse recommendation. An ideal way for recommendation is that to each user has assigned his/her most suitable recommendation algorithm [1]. In this way, the recommendation results will be closer to his/her real taste. However, in current recommendation design, all the users are forced to use the same algo-

rithms. To solve this problem, we proposed a personalized recommendation algorithm based on the hybrid method in [17]. Specifically, we assign to each user a parameter for the initial configuration setting and a parameter for the hybridization. We find that each user has his/her optimal parameters which are very different from user to user. We finally design a simple method to estimate users' personalized parameters and the recommendation accuracy can be improved accordingly.

2. Datasets and metrics

To test performance of our algorithm, we use two benchmark datasets. The MovieLens (<http://www.grouplens.org/>) data contains real data of 100,000 ratings from 943 users on 1,682 movies. The level of rating ranges from 1 to 5, as from worst to best. For the recommendation purpose, we are doing filtering process by considering the link with the rating more than 3. After the coarse gaining process, the data contains 82,520 user-object pairs including 943 users and 1,682 items. The Netflix data is random sampling of the whole records of user interaction in the Netflix website (<http://www.netflixprize.com>). It has 10,000 users, 6,000 movies, and 824,802 user-movie pairs. After the same link filtering like MovieLens data, there are 701,947 links left. To test the recommendation algorithm performance, the data is randomly divided into two parts: the training set E_T containing 90 percent of the data and the probe set E_P containing 10 percent of the data. The training set is treated as known information, while no information in probe set is allowed to be used for the training.

The recommendation algorithms should consider accuracy as the most important aspect for evaluating performance. A good algorithm should provide accurate recommendations which means each user can find what they like in the top of an ordered queue of all its uncollected objects. Here, Rank Score [16, 22] is to measure the ability of a recommendation algorithm to build a good ordering of items that matches the user's preference. For a specific user, the recommendation system can produce a ranking list of all his uncollected objects. We measure the rank of each user-items link in a probe set in the ranking list of this user. For example, an active user u_a who has 1,000 uncollected items, and the item i_b is at 10th place from the top of the list, so the place of this item is $RS_{ab} = 10/1000$. We say the rank score of this item is 0.01. Averaging all links in the probe set, we obtain the overall ranking score RS which can be used to measure the recommendation algorithm's accuracy. Obviously, the smaller RS we can obtain, the better our proposed algorithm is

$$RS = \frac{1}{|E_P|} \sum_{ab \in E_P} RS_{ab}. \quad (1)$$

Since real users usually consider only the top part of the ordered recommendation list, we also adapt two practical accuracy metrics to consider the number of a user's links in the probe set contained in the top- N places, namely Precision and Recall [20]. For a specific user u_a , the precision of recommendation, $P_a(N)$, is defined as

$$P_a(N) = \frac{d_a(N)}{N}, \quad (2)$$

where $d_a(N)$ indicates the number of relevant items (namely the items collected by u_a in the probe set) in the top- N places of recommendation list. We obtain the mean precision $P(N)$ of the entire system by averaging all users' precision score. Besides Precision, Recall have similar function to evaluate accuracy of recommendation algorithm from other perspective. Given a user u_a , recall score $\text{RE}_a(N)$ is defined as

$$\text{RE}_a(N) = \frac{d_a(N)}{N_a}, \quad (3)$$

where $d_a(N)$ indicates the number of relevant items (namely the items collected by u_a in the probe set) and N_a is the number of items user a collects in the probe set. Averaging all users' Recall Score, we obtain the mean recall score $\text{RE}(N)$ of the entire system.

3. User heterogeneity and personalized parameter

A recommendation system can be described by a bipartite network $A(U, M)$, denoting the user set $U = u_1, u_2, u_3, \dots, u_n$, and the item set as $M = m_1, m_2, m_3, \dots, m_q$. $A_{n \times q}$ is the adjacency matrix, where the element $A_{i\alpha} = 1$ if user i has collected item α , and $A_{i\alpha} = 0$ otherwise.

There are many recommendation algorithms, the important task of a recommender system is to generate an ordering list of the specific user's uncollected items. In this paper, we mainly consider the mass diffusion algorithm (Mass), heat conduction (Heats) and the hybrid algorithms of these two algorithms (Hybrids). Firstly, we briefly introduce them.

In Mass, the algorithm works by assigning one unit of resource to each item denoted by the vector f (where f_α is the resource possessed by item M_α), and then the redistribution is represented by $\tilde{f} = Wf$, where W is the resource allocation matrix, and the key factor to take advantage of the diffusion processes, and $k_\beta = \sum_{l=1}^n a_{l\beta}$ and $k_j = \sum_{\gamma=1}^m a_{l\gamma}$ denote the degree of item β and user j , respectively. The recommendation list of uncollected items is sorted by descending order of f_α^i . In fact, the process of mass diffusion is equivalent to resource allocation, which is also a three-step random

walk process. For example, for a target user i , the mass diffusion process is shown in Fig. 1 (a). The recommendation list for user i is obtained by ranking all his/her uncollected items in decreasing order according to their amount of gathered resources after the diffusion

$$W_{\alpha\beta} = \frac{1}{k_{\beta}} \sum_{j=1}^n \frac{a_{j\alpha} a_{j\beta}}{k_j}. \tag{4}$$

Another algorithm is Heats algorithm, which works similarly to Mass algorithm. Heats algorithm follows heat diffusion across the user-item bipartite network. In this algorithm, the collected by users items are considered as high temperature resources, otherwise the cold sources are uncollected items. The higher temperature of the item, the higher score it has. The heat conduction process is represented by

$$W_{\alpha\beta} = \frac{1}{k_{\alpha}} \sum_{j=1}^n \frac{a_{j\alpha} a_{j\beta}}{k_j}. \tag{5}$$

Similarly, Heats also redistributes resources which can be also random walk process. But the difference between heat conduction process and mass diffusion is in diffusion process because heat conduction redistributes resource by averaging its temperature of nearest the neighborhood. The Heats process is presented in Fig. 1 (b).

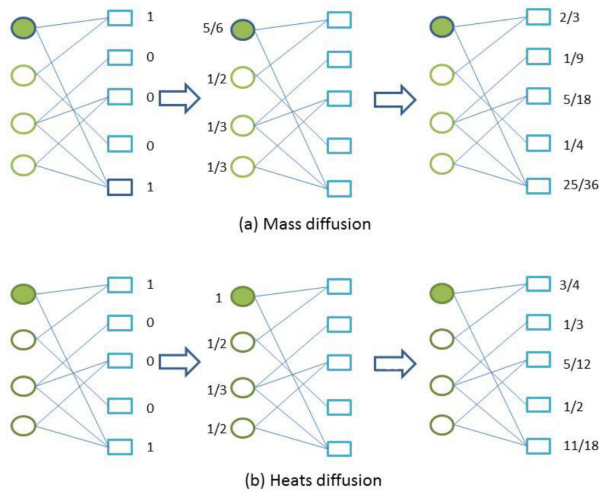


Fig. 1. (a) Mass algorithm and (b) Heats algorithm corresponding to Eqs. (4) and (5). The specific user is shown by the shaded circle, and objects are squares.

The originally Hybrids algorithm proposed in [17], combining Mass and Heats method by incorporating the hybridization parameter λ into the transition matrix normalization is

$$\tilde{f} = \lambda \frac{f^{\text{Mass}}}{\text{Max}(f^{\text{Mass}})} + (1 - \lambda) \frac{f^{\text{Heats}}}{\text{Max}(f^{\text{Heats}})}, \quad (6)$$

where the parameter $\lambda = 0$ gives the fully Heats algorithm and $\lambda = 1$ gives the Mass algorithm. When λ increases from 0 to 1, the Hybrids algorithm will change from Heats to Mass. This Hybrid algorithm was shown to be an effective way to solve the accuracy–diversity dilemma [24]. With the hybrid parameter, the Hybrids algorithm can be chosen whether to recommend more popular items or unpopular items.

A heterogeneous initial resource distribution Mass algorithm also proposed in [24] which identifies the initial resource of item i is proportional to k_i^θ . So the initial resource matrix of this method can be presented as $f_\alpha = a_{ij}k_i^\theta$, where parameter θ is a negative number. It shows that this method can improve the accuracy of the recommendation compared to the Mass diffusion process.

In Hybrids method and heterogeneous initial resource distribution Mass algorithm, all the users are assumed to have the same parameter λ as in Hybrids algorithm and the same θ in heterogeneous initial resource distribution Mass algorithm. However, this assumption is not true in real case since some users may prefer popular items, while others may like the high quality items more. Thus, we apply the hybrid method with heterogeneous initial resource distribution method to the individual level. For example, each user can adjust his/her own *personalized hybrid parameter* λ_a and *personalized initial resource parameter* θ_b for $W_{\alpha\beta}$ to get the optimal recommendation for him/her. The optimal λ_a and θ of user i can be assigned when the rank score is minimized. Table I shows the recommendation metrics results of rank score, precision, and recall when each user is obtained with the optimal λ_i^* and θ_i (which refer to personalized hybrid parameter and personalized initial resource parameter).

The results in Table I indicate that the recommendation accuracy can be significantly improved. We find the optimal personalized parameter algorithm performs better in all metrics mentioned. Take MovieLens for example, with comparative performance of OCoHybrids method on Rank Score RS, metrics including Precision $P(50)$ and Recall $R(50)$ with the CoHybrids method can have an enhancement of 9.01%, 4.09% and 7.9%. As for Netflix dataset, OCoHybrids method also outperforms in terms of RS, $P(50)$ and $R(50)$ the previous CoHybrids method with enhancement of 10.8%, 0.9% and 2.8%, respectively.

TABLE I

Optimal algorithm performance for MovieLens data and Netflix data, in terms of Precision and Recall are corresponding to $L = 50$. HMass algorithm is Mass distribution process with heterogeneous initial resource. OMass algorithm refers to Mass algorithm Heat conduction process method. OHybrid refers to Hybrids algorithm with personalized λ_i for each user. CoHybrids algorithm refers to Hybrids algorithm with overall optimal initial resource parameter θ and overall optimal λ . OCoHybrids algorithm is improved CoHybrids method combining the optimal personalized initial resource parameter θ_i and optimal personalized λ_i . The parameters (ranging in the interval $[0, 1]$ for λ with step 0.05, and $[-5, 5]$ for initial resource parameter with step 0.1, for MovieLens data; optimal $\theta = -0.8$ in HMass, CoHybrids method, optimal $\lambda = 0.45$ in Hybrids and CoHybrids). Each number is obtained by averaging over 10 runs with independently random division of training set and probe set.

	Algorithm	RS	$P(50)$	$R(50)$
MovieLens	HMass	0.091843	0.074889	0.503311
	OMass	0.086254	0.0772	0.526511
	Hybrids	0.083672	0.077837	0.51726
	OHybrids	0.077818	0.079576	0.534584
	CoHybrids	0.08084	0.07807	0.510147
	OCoHybrids	0.073556	0.081379	0.554477
Netflix	HMass	0.051843	0.053099	0.432153
	OMass	0.045742	0.054357	0.453447
	Hybrids	0.050999	0.052088	0.431291
	OHybrids	0.047065	0.05224	0.431695
	CoHybrids	0.047842	0.053183	0.433328
	OCoHybrids	0.04266	0.0537	0.446098

From the above ideal situation of our algorithm, we want to first consider the property of users' optimal personalized hybrid parameter λ_i^* . In Fig. 2, we show the distribution of λ_i^* in MovieLens and Netflix. In MovieLens, there is one peak close to $\lambda_i^* = 0.5$, while other λ_i^* spread over the value from 0 to 1. In Netflix, there is also an obvious peak in $\lambda_i^* = 0.9$ and other λ_i^* are distributed between 0 and 1. These results indicate that users have quite different personalized hybrid parameters in real systems. Secondly, we further move to investigate the property of users' optimal initial resource parameter θ . We also find out the difference of users' optimal initial resource parameter. So if we use the same hybrid parameter and initial resource parameter for all users, many users cannot receive the best recommendations.

In order to design a method to predict users' two optimal personalized parameters (refer to initial resource and hybrid), we study the correlation of λ_i^* and θ_i with each users' ranking score. We found that the users' per-

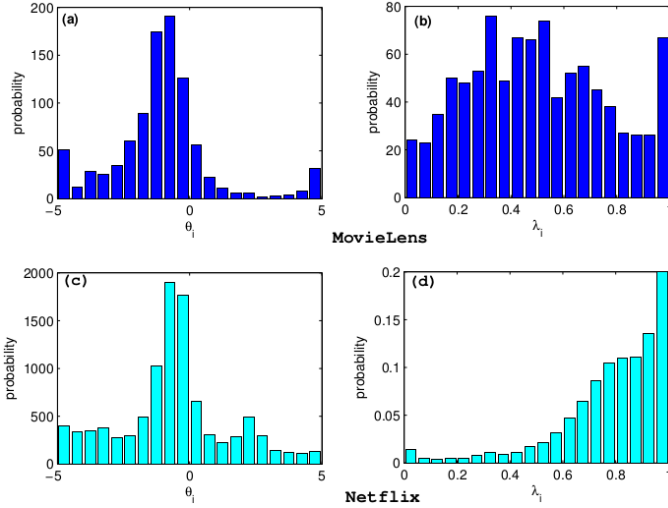


Fig. 2. The first two MovieLens subfigures show the distribution of θ_i and λ_i . Insets of two Netflix subfigures also show the distribution of θ_i and λ_i .

sonalized parameters are different, but their taste is more or less lasting in the long period. We design a new method to detect predicted users' optimal personalized parameters. In Section 2, we divided dataset into two parts: a training set and probe set. Because the probe set is containing an unknown link for test the algorithm performance, the training set can be used to identify personalized parameters. Specifically, we divided the training set into two parts which are corresponding to T-training set and T-probe set. The ration of these two parts is 9 : 1. By tuning these T-training set and T-probe set, it is found the optimal personalized parameter for this dividing data set (refer to original training set). In order to predict users' personalized parameters, it is useful to normalize noise of dataset by dividing more times to have normalized personalized parameters which are closer to optimal personalized parameter. We choose dividing the training set 100 times to get 100 times personalized parameters for each user depending on each user's minimal ranking score each time.

We also consider each user having several ranking scores which are very similar. Thus, normalizing the personalized parameters is necessary. As expected, the personalized λ_i and personalized θ_i we obtained are similar to the optimal personalized parameters. Therefore, in the next section, we will propose a strategy to assign to each user a suitable personalized hybrid parameter λ_i and personalized initial resource parameter θ_i based on tuning training set.

4. Recommendation with personalized algorithm

According to the analysis above, we propose a personalized initial resource and hybrid algorithm (PIHP) as

$$\tilde{f}_i = \lambda_i \frac{f^{\text{Mass}}}{\text{Max}(f^{\text{Mass}})} + (1 - \lambda_i) \frac{f^{\text{Heats}}}{\text{Max}(f^{\text{Heats}})}, \quad (7)$$

where λ_i is the users' personalized hybrid parameter collected by the user's minimal ranking score. To improve the feasibility of the method, PIHP obtained the λ_i only by training set. We divide the original training set into new training set and probe set 100 times. The ratio of these two sets is also 9 : 1. We consider the heterogeneous initial resource distribution [24] which can identify the initial resource of item i proportional to k_i^θ . This parameter also can be personalized by the same method as personalized hybrid parameter obtained method. Figures 3 (a), (c) show the positive correlation of optimal θ_i and personalized θ_i in two benchmark data MovieLens and Netflix, and Figs. 3 (b), (d) illustrate the correlation of optimal λ_i and personalized λ_i .

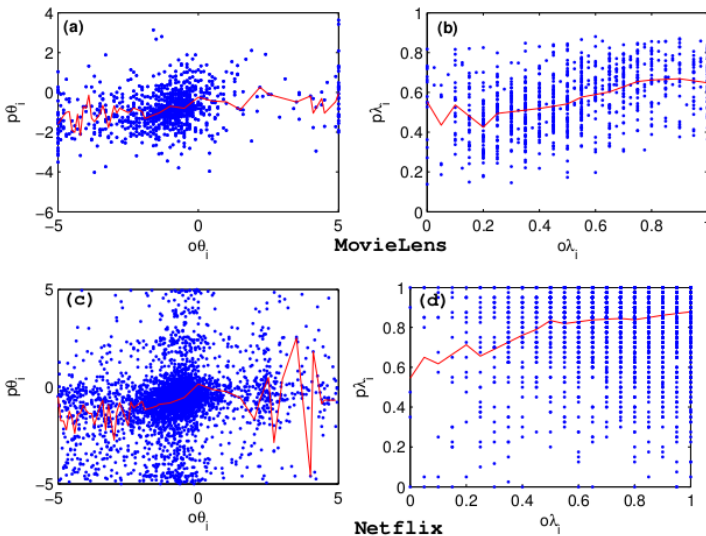


Fig. 3. The first two MovieLens subfigures show the correlation of optimal θ_i and personalized θ_i , and also for λ_i . Insets of two Netflix subfigures also show correlation with optimal parameters and personalized parameters, respectively.

The results for several algorithms are given in Table II. We find the PCoHybrids algorithm performs better in all metrics mentioned. Take MovieLens for example, with comparative performance on Rank Score RS, metrics

including Precision $P(50)$ and Recall $R(50)$ in the PCoHybrids method can have an enhancement of 2.07%, 3.3% and 4.5%. As for Netflix dataset, PCoHybrids method also outperforms in terms of RS, $P(20)$ and $R(20)$ the previous CoHybrids method with enhancement of 2.01%, 1.6% and 1.1%, respectively. Actually, the setting of personalized parameters is more significant for individuals' satisfactions and also the overall accuracy is still improved. In optimal personalized parameters situation, we still have some improving space to help people attain more satisfaction from personalized recommendation system.

TABLE II

Algorithm performance for MovieLens data and Netflix data, in terms of Precision, and Recall are corresponding to $L = 50$. The parameters ranging in the interval $[0, 1]$ for λ with step 0.05, and $[-5, 5]$ for initial resource parameter with step 0.1; for MovieLens data, optimal $\theta = -0.8$ in HMass, CoHybrids method, optimal $\lambda = 0.45$ in Hybrids and CoHybrids. Each number is obtained by averaging over 10 runs with independently random division of training set and probe set.

	Algorithm	RS	$P(50)$	$R(50)$
MovieLens	HMass	0.091843	0.074889	0.503311
	PMass	0.091328	0.075186	0.50431
	Hybrids	0.083672	0.077837	0.51726
	PHybrids	0.082098	0.078855	0.527373
	CoHybrids	0.08084	0.07807	0.510147
	PCoHybrids	0.079165	0.080742	0.534675
Netflix	HMass	0.051843	0.053099	0.432153
	PMass	0.050948	0.053134	0.431401
	Hybrids	0.050999	0.052088	0.431291
	PHybrids	0.050659	0.052448	0.434766
	CoHybrids	0.047842	0.053183	0.433328
	PCoHybrids	0.04688	0.054089	0.438163

5. Conclusion

Generally speaking, there is no optimal recommendation algorithm but most suitable recommendation algorithm for a user. Based on this, we proposed a personalized recommendation algorithm. Each user has assigned his/her personalized parameters according to his/her historical choices. We find that users personalized parameters are indeed different from each other. After introducing the personalized parameters, the recommendation accuracy is found to be improved.

In practice, an applicable and feasible way to implement our method is building an open recommender system, where users can help themselves to find their best experienced algorithm (or parameter). For example, the system can set a bar controlling the parameter of the algorithm on the website. By adjusting the parameter setting, the users can obtain popular (hot) or niche (novel) items. Gradually, the whole system will be guided to better and better state. In this sense, we argue that our work is of good contribution from practical point of view.

We thank Prof. Yi-Cheng Zhang and Prof. An Zeng for their help on this work. This work was partially supported by the EU FP7 Grant 611272 (project GROWTHCOM), NSFC (No. 11547040), the Science and Technology Innovation Commission of Shenzhen (No. JCYJ20150625101524056), the Education Fund of Guangdong Province (No. 2015KQNCX143), and the Tencent Open Research Fund. Correspondence and requests for materials should be addressed to H.L. and M.-H.L.

REFERENCES

- [1] L. Lü *et al.*, *Phys. Rep.* **519**, 1 (2012).
- [2] D. Goldberg, D. Nichols, B.M. Oki, D. Terry, *Commun. ACM* **35**, 61 (1992).
- [3] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, *Lect. Notes Comput. Sci.* **4321**, 291 (2007).
- [4] T. Hofmann, *ACM T. Inform. Syst.* **22**, 89 (2004).
- [5] M.J. Pazzani, D. Billsus, *Lect. Notes Comput. Sci.* **4321**, 325 (2007).
- [6] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, *Inform. Retrieval* **4**, 133 (2001).
- [7] D.M. Blei, A.Y. Ng, M.I. Jordan, *J. Mach. Learn. Res.* **3**, 993 (2003).
- [8] C. de Kerchove, P. Van Dooren, *SIAM News* **43**, 1314 (2008).
- [9] Z.-K. Zhang, T. Zhou, Y.-C. Zhang, *J. Comput. Sci. Technol.* **26**, 767 (2011).
- [10] S. Maslov, Y.-C. Zhang, *Phys. Rev. Lett.* **87**, 248701 (2001).
- [11] Y.-C. Zhang, M. Blattner, Y.-K. Yu, *Phys. Rev. Lett.* **99**, 154301 (2007).
- [12] P. Laureti, L. Moret, Y.-C. Zhang, Y.-K. Yu, *Europhys. Lett.* **75**, 1006 (2006).
- [13] J. Ren, T. Zhou, Y.-C. Zhang, *Europhys. Lett.* **82**, 58007 (2008).
- [14] A. Stojmirovic, Y.-K. Yu, *J. Comput. Biol.* **14**, 1115 (2007).
- [15] H. Liao, G. Cimini, M. Medo, *Lect. Notes Comput. Sci.* **7661**, 421 (2012).
- [16] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, *Phys. Rev. E* **76**, 046115 (2007).
- [17] T. Zhou *et al.*, *Proc. Natl. Acad. Sci. USA* **107**, 4511 (2010).

- [18] L. Lü, W. Liu, *Phys. Rev. E* **83**, 066119 (2011).
- [19] J.-G. Liu, T. Zhou, Q. Guo, *Phys. Rev. E* **84**, 037101 (2011).
- [20] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, *ACM Trans. Inf. Syst.* **22**, 5 (2004).
- [21] F.-G. Zhang, A. Zeng, *Europhys. Lett.* **100**, 58005 (2012).
- [22] T. Qiu, G. Chen, Zi.-Ke. Zhang, T. Zhou, *Europhys. Lett.* **95**, 58003 (2011).
- [23] Q.-M. Zhang, A. Zeng, M.-S. Shang, *PLOS ONE* **10**, 1371 (2006).
- [24] A. Zeng, C.H. Yeung, M.-S. Shang, Y.-C. Zhang, *Europhys. Lett.* **97**, 18005 (2012).