Improving information filtering via network manipulation

Fuguo Zhang[1,2] and An Zeng[2(a)]

[1] *School of Information and Technology, Jiangxi University of Finance and Economics*
Nanchang 330013, PRC
[2] *Department of Physics, University of Fribourg, Chemin du Musée 3 - CH-1700 Fribourg, Switzerland*

Abstract – The recommender system is a very promising way to address the problem of overabundant information for online users. Although the information filtering for the online commercial systems has received much attention recently, almost all of the previous works are dedicated to design new algorithms and consider the user-item bipartite networks as given and constant information. However, many problems for recommender systems such as the cold-start problem (*i.e.*, low recommendation accuracy for the small-degree items) are actually due to the limitation of the underlying user-item bipartite networks. In this letter, we propose a strategy to enhance the performance of the already existing recommendation algorithms by directly manipulating the user-item bipartite networks, namely adding some virtual connections to the networks. Numerical analyses on two benchmark data sets, *MovieLens* and *Netflix*, show that our method can remarkably improves the recommendation performance. Specifically, it not only improves the recommendations accuracy (especially for the small-degree items), but also helps the recommender systems generate more diverse and novel recommendations.

Introduction. – In the Internet era, the rapid growth of the World Wide Web leads to a serious problem of information overload: people are now facing too many choices to be able to find out the most relevant ones [1]. So far, the most promising way to efficiently filter the abundant information is to employ the personalized recommendations [2,3]. That is to say, using the personal history record of a user to uncover his preference and to return each user the most relevant items according to his taste [4]. For instances, `youtube.com` uses people's video viewing record to provide individual suggestions for their potential interested videos.

There are already many recommendation algorithms for the online user-item commercial systems. Among these algorithms, the simplest ones are the popularity-based recommendations, which recommend the most popular items to users. However, such recommendations are not personalized so that identical items are recommended to individuals with far different tastes. By comparison, the collaborative filtering makes use of collective data from individual preferences to provide personalized recommendations [5,6]. Recently, recommendation algorithms have been proposed from a physics perspective. For example, the process of mass diffusion (MD) was applied on the user-item bipartite networks to explore items of potential interest for a user [7]. The mass diffusion algorithm outperforms the previous ones in the recommendation accuracy. However, such method is still biased to popular items even if individual preferences are considered. An alternative approach, based on the heat conduction (HC) on the user-item graphs, was thus introduced [8]. This algorithm provides users with many novel items and leads to diverse recommendations among users. However, HC has low accuracy compared with MD. This drawback is eventually solved by combining MD with HC in a hybrid approach, which can be well tuned to obtain significant improvement in both recommendation accuracy and item diversity [9]. More recently, the long-term influence of such hybrid approach on network evolution has been studied [10].

However, all these methods focus on improving the recommendation from the system point of view. The recommendation on the items with little information is actually still a critical challenge [4]. For the fresh or unpopular items (also called niche items), it is very difficult to predict the potential users who are going to be interested in

[(a)] E-mail: `an.zeng@unifr.ch`

them due to the lack of historical records. Such problem is always referred as item cold-start problem and many researches have been dedicated to solve this problem. Related works concerning this issue are mainly based on modifying the existing methods by introducing some parameters [11–13].

Different from previous works, we tried to solve the item cold-start problem through a very fundamental way in this letter. Instead of designing a new recommendation algorithm, we solve the item cold-start problem by directly manipulating the underlying user-item bipartite networks [14,15]. Actually, the idea of the network manipulation has been applied to enhance many kinds of network functions such as synchronization [16,17], traffic dynamics [18], percolation [19,20], navigation [21] and so on. In our case, we first analyze the historical record of each item and accordingly add some virtual connections to the networks (especially for the small-degree items) to provide the recommendation algorithm with more information. By using the MD and the hybrid algorithms, we find that the recommendation accuracy for the small-degree items can be largely enhanced after manipulating the networks. The further test on the overall recommendation metrics and our method are shown to help the MD and the hybrid algorithms to significantly improve the recommendation diversity.

Recommendation algorithms. – Online commercial systems can be well described by the user-item bipartite networks. If a user collects an item, a link is drawn between them. Specifically, we consider a system of N users and M items represented by a bipartite network with adjacency matrix A, where the element $a_{i\alpha}=1$ if a user i has collected an item α, and $a_{i\alpha}=0$ otherwise (throughout this paper we use Greek and Latin letters, respectively, for item- and user-related indices).

There are many recommendation algorithms. In this letter, we mainly consider the mass diffusion (MD), heat conduction (HC) and the corresponding hybrid algorithms of these two algorithms (Hybrid). We first briefly describe these algorithms.

For a target user i, the MD algorithm [7] starts by assigning one unit of resource to each item collected by i, and redistributes the resource through the user-item network. We denote the vector $\mathbf{f}^i$ as the initial resources on items, where the α-th component f^i_α is the resource possessed by item α. Recommendations for the user i are obtained by setting the elements in $\mathbf{f}^i$ to be $f^i_\alpha=a_{i\alpha}$, in accordance with the items the user has already collected.

The redistribution is represented by $\widetilde{\mathbf{f}^i}=W\mathbf{f}^i$, where

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

is the diffusion matrix, with $k_\beta=\sum_{l=1}^{N}a_{l\beta}$ and $k_j=\sum_{\gamma=1}^{M}a_{j\gamma}$ denoting the degree of item β and user j, respectively. The resulting recommendation list of uncollected items is then sorted according to $\widetilde{f}^i_\alpha$ in descending order. Physically, the diffusion is equivalent to a three-step random walk starting with k_i units of resources on the target user i and the process can be seen in fig. 1(a). The *recommendation score* of an item is taken to be its amount of gathered resources after the diffusion. This algorithm was shown to enjoy a high recommendation accuracy.

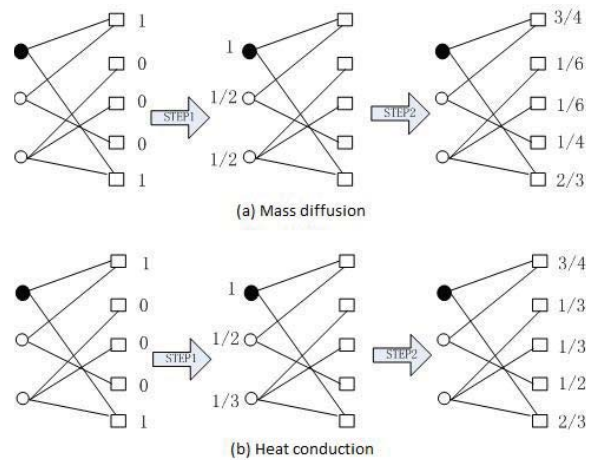

Fig. 1: The illustration of the (a) MD and (b) HC algorithms on the bipartite user-object network. Users are shown as circles; objects are squares. The target user is indicated by the shaded circle.

The HC algorithm [8] works similarly to the MD algorithm, but instead it follows a conductive process represented by

$$W_{\alpha\beta}=\frac{1}{k_\alpha}\sum_{j=1}^{N}\frac{a_{j\alpha}a_{j\beta}}{k_j}. \tag{2}$$

Physically, the recommendation scores can be interpreted as the temperature of an item, which is the average temperature of its nearest neighborhood, *i.e.*, of its connected users. The higher the temperature of an item, the higher its recommendation score. The HC process can be seen in fig. 1(b). By using this algorithm, the items with small degree can receive a relatively high recommendation score and finally be promoted to appear in the top recommendation list.

The Hybrid algorithm of MD and HC was proposed in [9], with the new recommendation score $\widetilde{h}_\alpha$ given by

$$\widetilde{h}_\alpha=\lambda\frac{\widetilde{f}^{\mathrm{MD}}_\alpha}{\mathrm{Max}(\widetilde{f}^{\mathrm{MD}})}+(1-\lambda)\frac{\widetilde{f}^{\mathrm{HC}}_\alpha}{\mathrm{Max}(\widetilde{f}^{\mathrm{HC}})}, \tag{3}$$

where the parameter λ adjusts the relative weight between the two algorithms. When λ increases from 0 to 1, the Hybrid algorithm changes gradually from HC to MD. Such hybrid approach was shown to achieve both accurate and diverse recommendation.

Data. – In order to test the performance of the recommendation results, we use two benchmark data sets in this letter. The first one is the MovieLens data [22] which has 1682 movies (items) and 943 users. The other is Netflix data [23] consisting of 10000 users and 6000 movies. The data sets are random samplings of users activity records in these two online systems. In both data sets, users can vote movies by giving different rating levels from 1 to 5 (*i.e.*, worst to best). Here, only the ratings larger than 2 are considered as a link. After this preliminary filtering, there are finally 82520 links in MovieLens data and 701947 links in the Netflix data. Each data is then randomly divided into two parts: the training set (E^T) and the probe set (E^P). The training set contains 90% of the original data and the recommendation algorithm runs on it. The probe set has the remaining 10% of the data and will be used to test the performance of the recommendation results. In our simulation, we actually try several different divisions of the E^T and E^P, and the results are quite robust.

The network manipulating method. – The network manipulating (NM) method takes place after dividing the data into E^P and E^T. The main idea of NM is to add some virtual links to the training set E^T, so that the niche items will have more information to provide to the recommendation algorithm. Denoting Q as the fraction of links added, the total number of virtual links will be $Q|E^T|$. For each item, the probability to receive virtual links is related to its degree, *i.e.*, $p_\alpha \propto k_\alpha^{-\delta}$, where δ is a tunable parameter. When $\delta > 0$, the items with smaller degree tend to receive more links, and vice versa. Supposing an item α is selected to receive a link, the virtual link will connect to the user who enjoys the highest average similarity to the already existing selectors of the item α. In this letter, the similarity is calculated by the *Salton Index* [24] as

$$s_{ij} = \frac{|\Gamma(i) \cap \Gamma(j)|}{\sqrt{k_i k_j}}, \tag{4}$$

where $\Gamma(i)$ denotes the set of neighbors of user i. After adding virtual connections to the networks, we will employ the MD revision and the Hybrid algorithms to do the recommendation and the combinations are denoted as "MD with NM" and "Hybrid with NM", respectively.

In the NM method, the added virtual links may happen to be one of the links in E^P. After running the recommendation process, the mass diffusion and Hybrid algorithms will generate the final recommendation list by sorting the recommendation score of all the uncollected items for the target user. In our simulation, if there is a virtual link connecting the target user and an item, this item is still considered as an uncollected item for the target user. Therefore, the NM method does not reduce the number of candidate items for recommendation (*i.e.*, the ratio of probe-set links out of all the non-existing links stays unchanged).

We remark that the NM method is quite efficient in sparse networks. For each item, the NM method has to calculate the average Salton similarity between the users who already selected it and the users who have not selected it. Since calculating the Salton similarity between users costs $O(\bar{k_i})$, the computational complexity for the NM procedure is $O(M\bar{k_\alpha}(N-\bar{k_\alpha})\bar{k_i})$. Since $\bar{k_\alpha} \ll N$, it can be further written as $O(M\bar{k_\alpha}(N-\bar{k_\alpha})\bar{k_i}) \approx O(M\bar{k_\alpha}N\bar{k_i}) = O((M*E*N*E)/(M*N)) = O(E^2)$, where E is the total number of links in the network.

Metrics for recommendation. – An effective recommendation should be able to accurately find the items that users like. In order to measure the recommendation accuracy, we make use of the *ranking score* (R). Specifically, R measures whether the ordering of the items in the recommendation list matches the users' real preference. As discussed above, the recommender system will provide each user with a ranking list which contains all his uncollected items. For a target user i, we calculate the position for each of his link in the probe set. Supposing one of his uncollected items α is ranked at the 5th place and the total number of his uncollected items is 100, the ranking score $R_{i\alpha}$ will be 0.05. In a good recommendation, the items in the probe set should be ranked higher, so that R will be smaller. Therefore, the mean value of the R over all the user-item relations in the probe set can be used to evaluate the recommendation accuracy as

$$R = \frac{1}{|E^P|} \sum_{i\alpha \in E^P} R_{i\alpha}. \tag{5}$$

The smaller the value of R, the higher the recommendation accuracy.

In reality, online systems only present users with only the top part of the recommendation list. Therefore, we further consider another more practical recommendation accuracy measurement called *precision*, which only takes into account each user's top-L items in the recommendation list. For each user i, his precision of recommendation is calculated as

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

where $d_i(L)$ represents the number of user i's deleted links contained in the top-L places in the recommendation list. For the whole system, the precision $P(L)$ can be obtained by averaging the individual precisions over all users with at least one link in the probe set.

Predicting what a user likes from the list of best sellers is generally easy in recommendations, while uncovering the users' very personalized preferences (*i.e.*, uncovering the unpopular items in the probe set) is much more difficult and important. Therefore, diversity should be considered as another significant aspect for recommender systems besides accuracy. In this letter, we employ two kinds of diversity measurement: *interdiversity* and *novelty.*

The interdiversity mainly consider how users' recommendation lists are different from each other. Here, we measure it by the Hamming distance. Denoting $C_{ij}(L)$ as the number of common items in the top-L place of

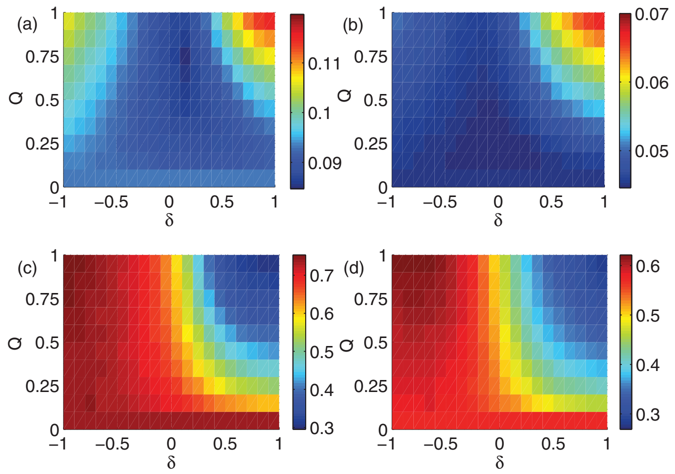

Fig. 2: (Color online) The overall ranking score R under different δ and Q in (a) MovieLens and (b) Netflix, and the local ranking score $R_{k\leqslant5}$ under different δ and Q in (c) MovieLens and (d) Netflix.

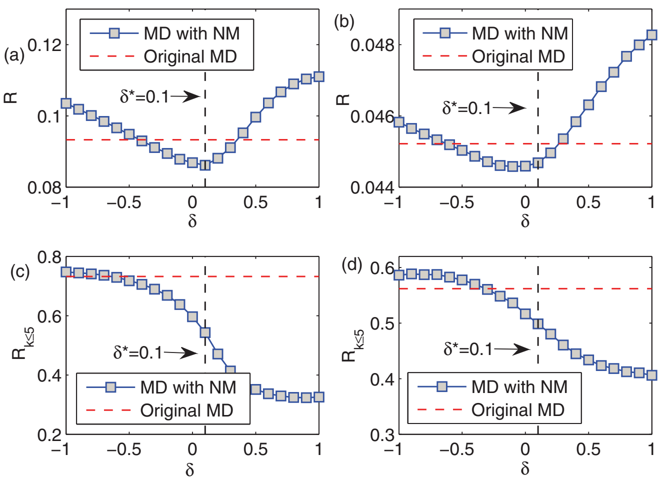

Fig. 3: (Color online) The overall ranking score R and the local ranking score $R_{k\leqslant5}$ of the MD with NM method under different δ when $Q=0.8$ in MovieLens ((a), (c)) and $Q=0.2$ in Netflix ((b), (d)). The vertical dashed line is the optimal δ we used.

the recommendation list of users i and j, their Hamming distance can be calculated as

$$H_{ij}(L)=1-\frac{C_{ij}(L)}{L}. \qquad (7)$$

Clearly, $H_{ij}(L)$ is between 0 and 1, which are, respectively, corresponding to the cases where i and j have the same or entirely different recommendation lists. Again, averaging $H_{ij}(L)$ over all pairs of users, we obtain the mean Hamming distance $H(L)$. A more personalized recommendation results in a higher $H(L)$.

The novelty measures the average degree of the items in the recommendation list. For those popular items, users may already get them from other channels. However, it is hard for the users to find the relevant but unpopular item. Therefore, a good recommender system should prefer to recommend small-degree items. The metric *novelty* can be expressed as

$$N_i(L)=\frac{1}{L}\sum_{\alpha\in O^i}k_\alpha, \qquad (8)$$

where O^i represents the recommendation list for user i. A low mean popularity $N(L)$ for the whole system indicates a high novel and unexpected recommendation of items.

Results. – We will begin our analysis with comparing the "MD with NM" and the original MD algorithm. We first investigate the result of the ranking score R under different δ and Q. Since the NM method partially aims at solving the item cold-start problem, we also define a local ranking score which is the average ranking score of the items with degree not larger than 5 (denoted as $R_{k\leqslant5}$). The results on MovieLens and Netflix data are reported in fig. 2. Clearly, with more links added to the network, $R_{k\leqslant5}$ becomes smaller. Consequently, the overall R is improved. Given the value of Q, a smaller δ yields a lower $R_{k\leqslant5}$, which means the recommendation for the small degree items becomes more accurate. However, the overall R does not monotonously change with δ. For each Q, there is a corresponding optimal δ which yields the best R.

For solving the item cold-start problem, figs. 2(c) and (d) suggest that a large Q generally works better. However, in order to keep a reasonable computational time of the recommendation algorithm, we select $Q=0.8$ in MovieLens data and $Q=0.2$ in Netflix data. The results of the ranking score are studied more detailedly in fig. 3. In this figure, besides the curve of the MD with NM, we plot the results of the original MD (without adding any virtual links) as a comparison. For overall R, δ near 0 clearly works better. However, positive δ is beneficial for improving the ranking score for small-degree items. In our simulation, we find that $\delta^*=0.1$ is the best trade-off between R and $R_{k\leqslant5}$. As we can see from fig. 3, under this δ^*, the overall R can outperform the original MD algorithm in both data sets. Moreover, the MD with MN method enjoys a lower $R_{k\leqslant5}$ than the original MD algorithm in both data sets. For the value of each metric, see table 1.

To show how the ranking score varies on items with different value of degrees, we additionally investigate an item-degree–dependent ranking score R_k [25]. R_k is defined as the average ranking score over items with the same value of degrees. In fig. 4, the relation between R_k and the item degree k is displayed, respectively, for the MovieLens and Netflix at the optimal parameters $\delta^*=0.1$. Besides the MD with NM method, we also plot the results of original MD for comparison. Obviously, the ranking score of small-degree items can be significantly improved

Table 1: The performance of the original MD and the MD with NM methods in *MovieLens* and *Netflix* data. The recommendation list length is set as $L = 20$. In the MD with NM method, the parameters are $Q = 0.8$, $\delta = 0.1$ in MovieLens and $Q = 0.2$, $\delta = 0.1$ in Netflix. The entries corresponding to the best performance over all methods are emphasized in black.

Network	Method	R	$R_{k\leqslant 5}$	$P(20)$	$H(20)$	$N(20)$
MovieLens	Original MD	0.0933	0.7324	0.1427	0.7161	303.8
	MD with NM	**0.0861**	**0.5430**	**0.1451**	**0.7437**	**293.3**
Netflix	Original MD	0.0452	0.5695	0.0808	0.5470	2841.6
	MD with NM	**0.0446**	**0.4986**	**0.0821**	**0.5618**	**2829.8**

Table 2: The performance of the original Hybrid and the Hybrid with NM methods in *MovieLens* and *Netflix* data. The recommendation list length is set as $L = 20$. In the Hybrid with NM method, the parameters are $Q = 0.1$, $\delta = 0.1$ in MovieLens and $Q = 0.1$, $\delta = 0.1$ in Netflix. The entries corresponding to the best performance over all methods are emphasized in black.

Network	Method	R	$R_{k\leqslant 5}$	$P(20)$	$H(20)$	$N(20)$
MovieLens	Original Hybrid	0.0759	0.5059	0.1532	0.8055	276.7
	Hybrid with NM	**0.0755**	**0.4898**	**0.1549**	**0.8173**	**269.9**
Netflix	Original Hybrid	**0.0446**	0.5532	0.0810	0.5491	2839.1
	Hybrid with NM	0.0447	**0.5232**	**0.0816**	**0.5563**	**2834.7**

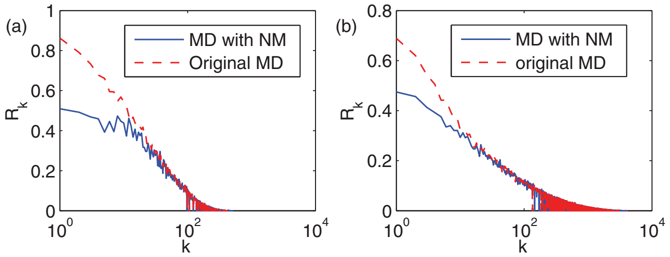

Fig. 4: (Color online) R_k *vs.* item degree k when using different methods in (a) MovieLens and (b) Netflix data sets. The parameters for the MD with NM method are $Q = 0.8$, $\delta = 0.1$ in MovieLens and $Q = 0.2$, $\delta = 0.1$ in Netflix.

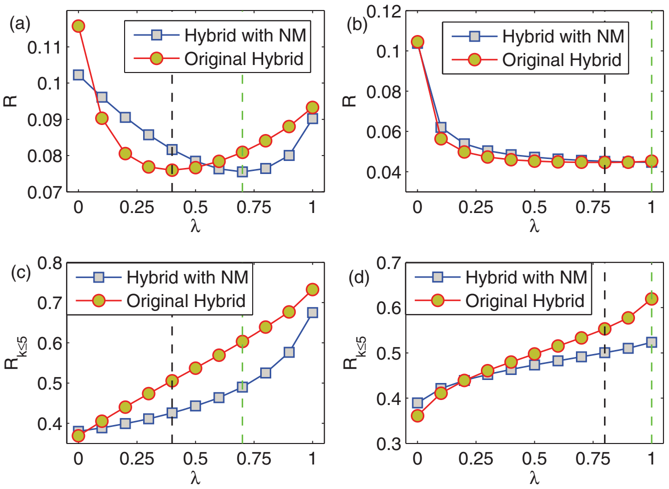

Fig. 5: (Color online) The overall ranking score R and the local ranking score $R_{k\leqslant 5}$ of the Hybrid with NM method under different λ when $Q = 0.1$, $\delta = 0.1$ in MovieLens ((a), (c)) and $Q = 0.1$, $\delta = 0.1$ in Netflix ((b), (d)). The black and green vertical dashed lines are the optimal λ^* for the original Hybrid method and the Hybrid with NM method, respectively.

by adding the virtual connections. Moreover, the ranking score of large-degree items can be effectively preserved.

As discuss above, another way to estimate the accuracy of the recommendation results is the precision. We select the recommendation list $L = 20$, and report the precision of MD with NM, the original MD algorithm in table 1. In addition to accuracy, the recommendation diversity is of great significance. For interdiversity, we can estimate how the recommendation results are different from user to user. A larger Hamming distance indicates a more personalized recommendation. Besides, the novelty is also an important aspect. With a small novelty, the average degree of the recommended items are low, so that more fresh items will appear in the recommendation list. Setting the recommendation list length as $L = 20$, the related results of different methods are reported in table 1. We can see that all these metrics are improved by the MD with NM method.

We then investigate the effect of NM method on the Hybrid algorithm. We first set $Q = 0.1$ and $\delta = 0.1$ in both data sets, the results are shown in fig. 5. As we can see, the minimum value of the Hybrid method stays almost unchanged after adding the virtual links. However, the $R_{k\leqslant 5}$ under the optimal λ^* in the Hybrid with NM method is lower than that in the original Hybrid method. The result indicates that the NM method can further solve the item cold-start problem in the Hybrid method without harming at all the overall recommendation accuracy. In our simulation, we try also large Q and δ, and the results show that the optimal R for the Hybrid algorithm becomes larger. Therefore, the Hybrid algorithm benefits from the NM method with relatively small Q and δ. Moreover, we find that the NM method can improve the recommendation diversity of the Hybrid algorithm. The detail value of each metric can be seen in table 2.

In the original Hybrid algorithm, the parameter λ can adjust the amount of score that the Hybrid algorithm gives to the small-degree items. When λ is close to 0, the HC algorithm has more weight in the Hybrid algorithm so that the small degree items will obtain more resource from the 3-step diffusion process. However, if a user cannot reach a certain item from the diffusion, the final resource of the item will be always 0 when λ is changed. In the NM method, the connectivity of the small-degree items can be effectively increased, so that some potential users who are interested in a small-degree item might be able to reach such item in the 3-step diffusion. Therefore, the NM method can further improve the recommendation accuracy of small-degree items in the Hybrid algorithm.

Conclusion. – Information abundance is a serious problem nowadays for online users. In order to filter irrelevant information, many recommendation algorithms have been proposed. In this field, one of the biggest challenges is the item cold-start problem, *i.e.*, the new items have too little historical record to be correctly recommended. So far, all the methods dedicated to solve this problem focused on modifying the existing methods by introducing some parameters. In this letter, we try to solve the problem by directly adding some virtual connection to the bipartite networks so that the niche items have enough information for the recommendation algorithms. Interestingly, besides improving the recommendation accuracy (especially for small-degree items), our method can enhance the recommendation diversity compared to the well-known Hybrid method of mass diffusion and heat conduction algorithms. In recommender systems, note that there are also the user cold-start problem and the more difficult cold-start problem for both users and items. These problems have to be addressed in other ways.

In practice, it is actually not necessary to add too many virtual links to the networks to solve the item cold-start problem. Generally, adding 10% links will be sufficient to further enhance the accuracy for those not so popular items in the Hybrid algorithm. Therefore, our method can be easily applied to real online commercial systems without increasing too much the computational complexity of the recommendation process. Finally, whether the current NM method is the optimal one for each recommendation algorithm is still unknown. For instance, some special algorithms such as the heat conduction algorithm, which mainly recommends niche items, might be required for a different virtual link adding strategy. Related problems ask for further investigation in the future.

* * *

We would like to thank Prof. Yi-Cheng Zhang for helpful suggestions. This work is partially supported by the Foundation of Jiangxi Provincial Department of Education (GJJ. 10696) and National Natural Science Foundation of China under Grant No. 71261009.

REFERENCES

[1] Broder A., Kumar R., Moghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A. and Wiener J., *Comput. Netw.*, **33** (2000) 309.

[2] Adomavicius G. and Tuzhilin A., *IEEE Trans. Know. Data Eng.*, **17** (2005) 734.

[3] Cacheda F., Carneiro V., Fernández D. and Formoso V., *ACM Trans. Web*, **5** (2011) 1.

[4] Lü L., Medo M, Yeung C. H., Zhang Y.-C., Zhang Z.-K. and Zhou T., *Phys. Rep.*, **519** (2012) 1, 10.1016/j.physrep.2012.02.006.

[5] Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R. and Riedl J., *Commun. ACM*, **40** (1997) 77.

[6] Herlocker J. L., Konstan J. A., Terveen K. and Riedl J. T., *ACM Trans. Inf. Syst. Secur.*, **22** (2004) 5.

[7] Zhou T., Ren J., Medo M. and Zhang Y.-C., *Phys. Rev. E*, **76** (2007) 046115.

[8] Zhang Y.-C., Blattner M. and Yu Y.-K., *Phys. Rev. Lett.*, **99** (2007) 154301.

[9] Zhou T., Kuscsik Z., Liu J.-G., Medo M., Wakeling J. R. and Zhang Y.-C., *Proc. Natl. Acad. Sci. U.S.A.*, **107** (2010) 4511.

[10] Zeng A., Yeung C. H., Shang M.-S. and Zhang Y.-C., *EPL*, **97** (2012) 18005.

[11] Lu L. and Liu W., *Phys. Rev. E*, **83** (2011) 066119.

[12] Qiu T., Chen G., Zhang Z.-K. and Zhou T., *EPL*, **95** (2011) 58003.

[13] Liu J.-G., Zhou T. and Guo Q., *Phys. Rev. E*, **84** (2011) 037101.

[14] Evans T. S. and Plato A. D. K., *Phys. Rev. E*, **75** (2007) 056101.

[15] Zhang C.-J. and Zeng A., *Physica A*, **391** (2012) 1822.

[16] Nishikawa T. and Motter A. E., *Proc. Natl. Acad. Sci. U.S.A.*, **107** (2010) 10342.

[17] Zeng A., Lu L. and Zhou T., *New J. Phys.*, **14** (2012) 083006.

[18] Yang H., Nie Y., Zeng A., Fan Y., Hu Y. and Di Z., *EPL*, **89** (2010) 58002.

[19] Schneider C. M., Moreira A. A., Andrade J. S. Jr., Havlin S. and Herrmann H. J., *Proc. Natl. Acad. Sci. U.S.A.*, **108** (2011) 3838.

[20] Zeng A. and Liu W., *Phys. Rev. E*, **85** (2012) 066130.

[21] Li G., Reis S. D. S., Moreira A. A., Havlin S., Stanley H. E. and Andrade J. S. jr., *Phys. Rev. Lett.*, **104** (2010) 018701.

[22] `http://www.grouplens.org/`.

[23] `http://www.netflixprize.com/`.

[24] Salton G. and McGill M. J., *Introduction to Modern Information Retrieval* (McGraw-Hill, Auckland) 1983.

[25] Zhou T., Jiang L.-L., Su R.-Q. and Zhang Y.-C., *EPL*, **81** (2008) 58004.