Information filtering by similarity-preferential diffusion processes

An Zeng, Alexandre Vidmer, Matúš Medo and Yi-Cheng Zhang[a]

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

PACS 89.75.-k – Complex systems
PACS 89.65.-s – Social and economic systems
PACS 89.20.Ff – Computer science and technology

Abstract – Recommender systems provide a promising way to address the information overload problem which is common in online systems. Based on past user preferences, a recommender system can find items that are likely to be relevant to a given user. Two classical physical processes, mass diffusion and heat conduction, have been used to design recommendation algorithms and a hybrid process based on them has been shown to provide accurate and diverse recommendation results. We modify both processes as well as their hybrid by introducing a parameter which can be used to enhance or suppress the weight of users who are most similar to the target user for whom the recommendation is done. Evaluation on two benchmark data sets demonstrates that both recommendation accuracy and diversity are improved for a wide range of parameter values. Threefold validation indicates that the achieved results are robust and the new recommendation methods are thus applicable in practice.

Introduction. – The amounts of data made available by modern World Wide Web sites far exceed the information capability of any individual. As the user base of these sites expands and increasing part of our lives is monitored and stored online, the problem of information overload becomes more and more acute. Information filtering tools thus have become essential to online users. Recommender systems represent an effective approach to information filtering where patterns in past user actions are analyzed to choose a small number of items that are likely to be appreciated by a given target user [1,2]. Approaches to recommendation include collaborative filtering [3], content-based analysis [4], dimensionality reduction techniques [5], tag-aware algorithms [6], trust-aware algorithms [7] and algorithms based on social impact [8]. An extensive review of various methods and their performance is presented in [9].

Information filtering has recently attracted the attention of physicists who have used network representations of the data, bipartite user-item networks in particular, to devise new recommendation algorithms motivated by heat conduction [10] and mass diffusion [11]. The hybrid of these two algorithms was shown to simultaneously improve recommendation accuracy and diversity [12]. This hybrid has been extended in multiple directions by modifying the initial configuration [13], introducing a ground node [14], and personalizing the hybridization parameter [15]. Network manipulation has been shown to solve the cold-start problem in recommendation [16]. A method for the removal of redundant links has been proposed to increase the efficiency of the recommendation process [17]. Preferential diffusion towards low-degree items has been shown to further enhance diversity and novelty of recommendations [18]. The issue of long-term influence of recommendation on the evolution of information systems has been recently studied [19]. See [20] for a review of recommender systems from a physicists' perspective.

Unlike the works above which alter the original hybrid method [12] by either manipulating the underlying bipartite network or by changing the hybridization procedure, we propose here to modify the diffusion process itself. In particular, we introduce an additional model parameter which makes it possible to increase the influence of users who are most similar to a target user for whom the recommendation is computed. We demonstrate that the resulting algorithm can improve recommendation accuracy and diversity, while increasing novelty of the recommended items (*i.e.*, decreasing their average degree). The optimal value of the newly introduced parameter strongly varies with data sparsity with sparse data requiring that lesser preference is given to similar users or, even, that their weight is made more equal to the weight of less similar

[a] E-mail: yi-cheng.zhang@unifr.ch

users. The parameter range over which the recommendation performance improves is broad which simplifies the new method's application in practice.

Recommendation algorithms. – Data produced by many e-commerce systems can be modeled by a bipartite network where users and objects are modeled by two distinct kinds of nodes and links represent the interactions of users with contents. We assume that there are I item nodes and U user nodes in total. A particular bipartite network can be represented by an adjacency matrix A, where the element $a_{i\alpha}$ equals 1 if user i has collected (or otherwise interacted with) item α, and 0 otherwise. For the sake of clarity, we label users and items with Latin and Greek letters, respectively. k_i and k_α thus denote the degree of user node i and item node α, respectively.

We first describe the similarity-preferential mass diffusion (SPMD) method. For a target user i for whom recommendation is done, each of the originally collected objects is assigned with one unit of resource. The resource of each object is then evenly distributed to all the neighboring users who have collected this object who receive $1/k_\alpha$ each. User j receives contribution from item α only if both i and j are connected to this item which is equivalent to $a_{i\alpha}a_{j\alpha} = 1$. The final resource received by user j can be thus written as

$$f_{ij}^{(SPMD)} = \sum_{\alpha=1}^{I} \frac{a_{i\alpha}a_{j\alpha}}{k_\alpha}. \tag{1}$$

The obtained f_{ij} value can be used to measure the similarity between users i and j (see, for example, [21]). In the original mass diffusion method [11], recommendation scores of distinct items are obtained by evenly distributing f_{ij} to all items connected with user j. As a result, the weight of user i is directly proportional to the similarity value f_{ij}.

To further boost the weight of the most similar users, we modify the similarity value from f_{ij} to f_{ij}^θ which is then evenly distributed to neighboring items as before. Here θ is a tunable parameter which results in the original mass diffusion (MD) method when $\theta = 1$, enhances the weight of similar users when $\theta > 1$, and suppresses the weight of similar users when $\theta < 1$ (note that $\theta < 0$ would lead to user weight decreasing with similarity which is not reasonable). The final amount of resource on item β is obtained by summing over all users who may send resource to it and therefore

$$f_{i\beta}^{(SPMD)} = \sum_{j=1}^{U} \frac{a_{j\beta}f_{ij}^\theta}{k_j} \tag{2}$$

The whole SPMD process is illustrated in fig. 1(a). To prevent recommendation of items that have been already collected by the target user i, we set $f_{i\beta}$ to zero $\forall\beta : a_{i\beta} = 1$. The recommendation list for this user is then obtained by sorting all items according to $f_{i\beta}$ in a descending order. From the community structure point of view, using

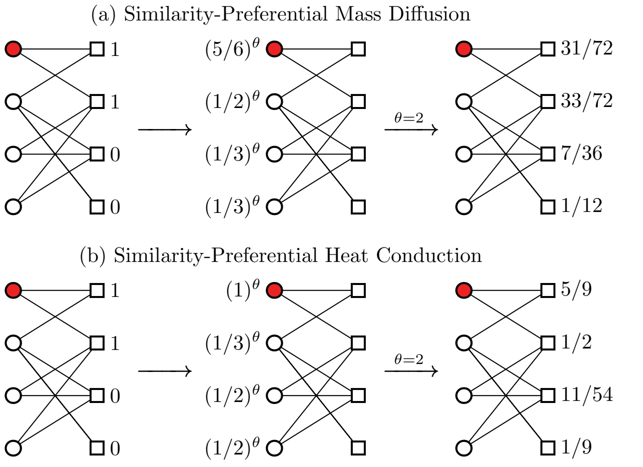

Fig. 1: (Color online) An illustration of two similarity-preferential processes: the initial configuration (left), the first step (middle) and the second step (right). Users and items are marked with circles and squares, respectively. Shaded circles indicate the target user for whom recommendation is done. To obtain the final resource values in the second step, $\theta = 2$ is used as an example.

$\theta > 1$ means that the target user is more likely to be recommended items collected by the users from i's community. This is in line with [22] which shows that clustering can emerge through common interests of users in online bipartite networks. A study of improving the link prediction performance by community structure information can be found in [23]. In addition to addressing a different problem (recommendation instead of link prediction), our approach is simpler because it does not require prior community analysis which is computationally expensive and often unreliable [24].

The similarity-preferential heat conduction (SPHC) method is similar to SPMD but follows a heat-conduction motivated formula for user similarity

$$f_{ij}^{(SPHC)} = \sum_{\alpha=1}^{I} \frac{a_{i\alpha}a_{j\alpha}}{k_j}. \tag{3}$$

In the second step, normalization is again with respect to the target node β instead of the initial node j, and thus

$$f_{i\beta}^{(SPHC)} = \sum_{j=1}^{U} \frac{a_{j\beta}f_{ij}^\theta}{k_\beta}. \tag{4}$$

The whole SPHC process is illustrated in fig. 1(b). When $\theta = 1$, this new method simplifies to the original heat conduction (HC) method [12].

Finally, we consider the same nonlinear hybridization of the two algorithms as in [12]. The user similarity becomes

$$f_{ij}^{(SPHY)} = \sum_{\alpha=1}^{I} \frac{a_{i\alpha}a_{j\alpha}}{k_\alpha^\lambda k_j^{1-\lambda}}. \tag{5}$$

and the final resource values are

$$f_{i\beta}^{(SPHY)} = \sum_{j=1}^{U} \frac{a_{j\beta} f_{ij}^{\theta}}{k_j^{\lambda} k_{\beta}^{1-\lambda}}. \tag{6}$$

The parameter $\lambda \in [0, 1]$ adjusts the relative weight of the two algorithms with $\lambda = 0$ and $\lambda = 1$ corresponding to SPHC and SPMD, respectively. When $\theta = 1$, the hybrid method of [12] is recovered (we label it as HY).

Data. – To evaluate the performance of the new methods, we use two benchmark data sets. The MovieLens data [25] is available at `http://www.grouplens.org/`. It contains 1682 movies and 943 users who rated the movies using the integer scale from 1 (worst) to 5 (best). The original data contains 10^5 ratings. To obtain an unweighted bipartite network, we represent all ratings of 3 or more as a link between a respective user and item. The resulting number of links is 82520. The Netflix data is a subset of the original data set released for the purpose of the NetflixPrize [26] and available at `http://www.netflixprize.com/`. The subset contains 3000 users and 3000 movies chosen from the original data at random and all links among them. Since the rating scale is the same as in MovieLens, we applied the same coarse-graining procedure and obtained 197248 links.

Evaluation. – We apply the standard evaluation procedure based on randomly dividing the data into two parts: a training set E^T containing 90% of all links (unless stated otherwise) and a probe set E^P containing the rest. Performance of a given algorithm is then assessed based on the position of hidden links from E^P in recommendation lists based only on the data in E^T. Results presented here are obtained by averaging over 10 independent E^P/E^T divisions. A more cautious evaluation based on threefold validation is presented at the end of the following section.

Ranking score (RS) measures the average rank of E^P entries in recommendation lists. If rank of item α in the recommendation list of user i is $r_{i\alpha}$, we say that its ranking score is $RS_{i\alpha} = r_{i\alpha}/I$. By averaging this quantity over all probe entries (i, α), we obtain the ranking score

$$RS = \frac{1}{|E^P|} \sum_{(i,\alpha)\in E^P} RS_{i\alpha}. \tag{7}$$

The lower the value, the better the recommendation method.

In real online systems, only the top part of a recommendation list is shown and even when users can usually inspect the rest, they rarely go far down the list [27]. We thus accompany the ranking score with another accuracy measure, *precision*, which considers only top-L items in each user's recommendation list. For user i, precision is computed as

$$P_i(L) = \frac{d_i(L)}{L} \tag{8}$$

where $d_i(L)$ is the number of i's probe set entries that are present in the top-L recommendation list (they correspond to successful recommendations). The precision $P(L)$ for the whole system can be obtained by averaging $P_i(L)$ over all users with at least one entry in the probe set. We use $L = 20$ in all L-dependent metrics presented here. While absolute results change with L, the relative comparison of methods is largely independent of L.

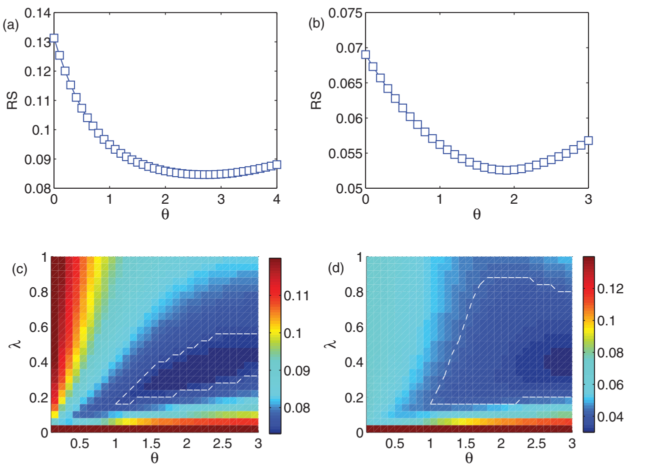

Fig. 2: (Color online) SPMD's ranking score *vs.* θ for MovieLens (a) and Netflix (b). SPHY's ranking score *vs.* θ and λ for MovieLens (c) and Netflix (d). In panels (c) and (d), the dashed line marks the region where RS is better than the RS value achievable with $\theta = 1$.

Along with accuracy, diversity of recommendations is also important for a recommender system [28]. We use two different diversity metrics here. *Personalization* measures the average difference between recommendation lists of distinct users. Denoting the number of items that occur in top L places of recommendation lists for both user i and user j as $C_{ij}(L)$, the Hamming distance of the lists is

$$H_{ij}(L) = 1 - \frac{C_{ij}(L)}{L}. \tag{9}$$

The extreme values of 0 and 1 correspond to identical and entirely dissimilar lists, respectively. By averaging $H_{ij}(L)$ over all user pairs to obtain $H(L)$, one obtains the personalization value of a recommendation method. The higher the value of $H(L)$, the more personalized the recommendation.

A good recommendation method should not place only high-degree items at the top of the resulting recommendation lists because those popular items are likely to be already known to the users and thus of little practical value. To this end, we measure recommendation *novelty* which focuses on the average degree of recommended items. Denoting the set of items in top L places of user i's list as O_L^i, novelty can be computed as

$$N(L) = \frac{1}{UL} \sum_{i=1}^{U} \sum_{\alpha\in O_L^i} k_{\alpha}. \tag{10}$$

The lower the value, the more novel the recommendation.

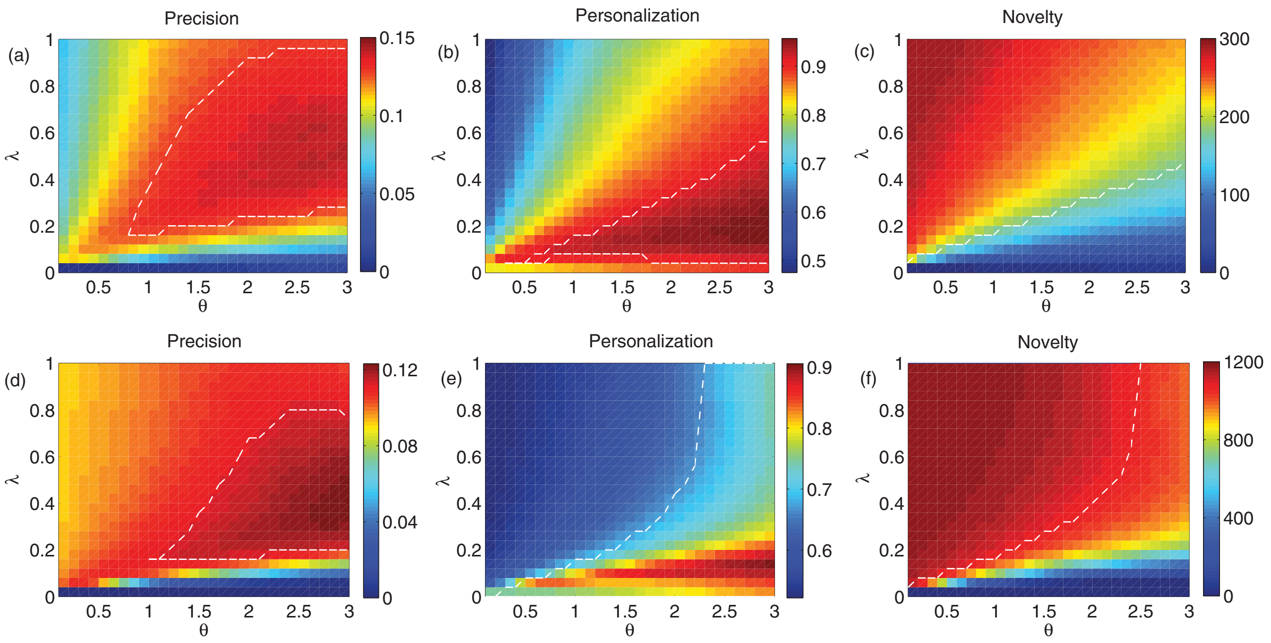

Fig. 3: (Color online) SPHY's precision, personalization and novelty *vs.* θ and λ for MovieLens (top) and Netflix (bottom). In each panel, the dashed line marks the region where the original HY method (with λ that optimizes the ranking score) is outperformed by the SPHY method.

Table 1: Results achieved by different recommendation methods. Parameter values (if any) were always set by maximizing the ranking score for each individual method. Standard deviation values obtained by independent training set-probe set divisions are negligible with respect to differences among the methods.

Data	Method	RS	$P(20)$	$H(20)$	$N(20)$
MovieLens	MD	0.096	0.115	0.703	278
	SPMD	0.084	0.129	0.837	238
	HC	0.135	0.006	0.862	7
	SPHC	0.122	0.013	0.904	14
	HY	0.075	0.129	0.903	179
	SPHY	0.072	0.141	0.917	174
Netflix	MD	0.056	0.099	0.551	1169
	SPMD	0.053	0.106	0.632	1116
	HC	0.113	0.000	0.763	1
	SPHC	0.108	0.001	0.830	2
	HY	0.052	0.112	0.667	1058
	SPHY	0.045	0.123	0.759	955

Results. – We first assess the one-parametric SPMD method. As shown in fig. 2(a), (b), the best ranking score is obtained with $\theta > 1$ for both data sets. The optimal value and relative improvement with respect to the original MD method is $\theta^* = 2.6$ and 12.4% (for Movielens) and $\theta^* = 1.9$ and 6.4% (for Netflix), respectively. SPHC can be studied in the same way but as mentioned in [12], recommendations obtained purely with heat diffusion are too peculiar to be actually useful (their accuracy is very low). The resulting values are nevertheless reported in table 1.

As can be seen in fig. 2(c), (d), the ranking score of the two-parametric SPHY method outperforms the best result achievable under $\theta = 1$ in a broad region. The optimal parameter values and the relative improvement with respect to the $\theta = 1$ case is $\theta^* = 2.0$, $\lambda^* = 0.32$ and 5.0% (for MovieLens) and $\theta^* = 3.0$, $\lambda^* = 0.40$ and 12.0% (for Netflix). Precision values achieved with SPHY for different parameter settings are shown in fig. 3 (left column). Using the optimal parameter values determined from the ranking score, relative precision improvement is 9.0% and 10.5% for Movielens and Netflix, respectively. These results confirm that introducing the similarity preference in the diffusion process can indeed improve the accuracy of recommendations.

Figure 3 (middle and right column) shows that the impact of θ on recommendation diversity is strictly monotonous: as θ grows, personalization increases and novelty improves. The increase of personalization with θ is a direct consequence of effectively relying on fewer users in producing recommendation for a target user. Since this small group can easily differ for one target user to another, the resulting recommendations are also different and thus personalization is high. The connection between θ and novelty is less direct but equally apparent. By using the optimal parameters determined from the ranking score, personalization and novelty are improved by 1.6% and 2.4% (for MovieLens) and 13.8% and 9.7% (for Netflix), respectively.

Table 1 reports detailed results for all four metrics and all recommendation methods. One can see that for both data sets and all metrics, the previously known methods

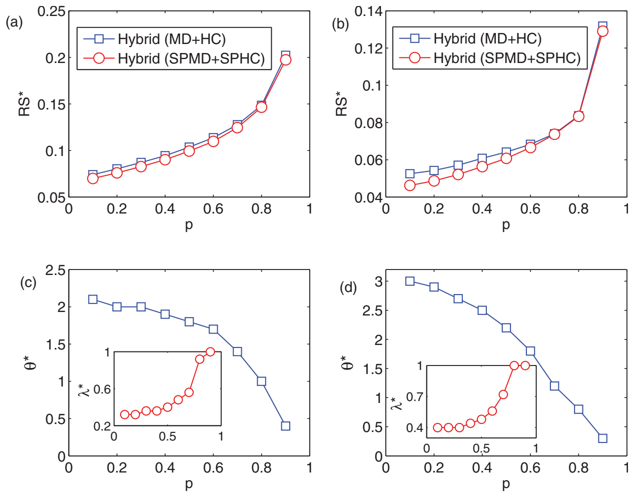

Fig. 4: (Color online) The optimal ranking score RS^* of HY and SPHY *vs.* p in MovieLens (a) and Netflix (b). SPHY's optimal parameter values *vs.* p in MovieLens (c) and Netflix (d).

(MD, HC, and HY) are outperformed by their variants employing similarity-preferential diffusion. Even in the case of HC, which is a widely known diversity-favoring method, SPHC achieves significantly higher personalization for both MovieLens and Netflix data. We also examine the performance of our methods in the `delicious.com` data which is very sparse and features a hierarchically organized social network. Compared to the results in the Movielens and Netflix data, the advantage of SPHY over HY is smaller in the Delicious data.

We further investigate the effect of data sparsity on the HY and SPHY method. To this end, we move fraction p of all links to the probe set and use the remaining $1-p$ links as the training set (the previously reported results are recovered by setting $p = 0.1$). As p increases (corresponding to sparser input data), the optimal ranking score naturally increases and the difference between HY and SPHY decreases (see fig. 4). The latter observation is no surprise as with very sparse data, information on user similarity is scarce and noisy and enhancing user weight based on this similarity thus yields smaller benefits. Interestingly, the optimal parameter values (again obtained by maximization of the ranking score) depend strongly on the data sparsity. In particular, as data gets very sparse ($p \gtrsim 0.8$ for both data sets), $\lambda^* = 1$ (corresponding to using only mass diffusion) and θ^* becomes smaller than one. Both observations are understandable: when the data is sparse, it is better to rely on popular items (hence SPMD instead of SPHY) and make the weight of all users more equal (with $\theta < 1$) as opposed to making it more unequal when the data is dense. In summary, there is a crucial difference in how recommendation methods behave on dense and sparse data sets.

As we have seen above, the parameter values under which SPHY yields the best ranking score strongly depend on the input data. This means that no universally applicable values exist and to employ the recommendation method in practice, one has to learn the optimal parameter values from the data. To mimic this process in evaluation of recommender system, one typically uses a so-called threefold validation where a small part (usually 10% of all data) is moved from the previously introduced testing set to a learning set E^L. By comparing the recommendations based on the remaining training set with the learning set, one can determine the optimal parameter values that optimize a given aspect of recommendation (here we stick with the previously used RS minimization as the only criterion). The learned optimal values are then used to produce recommendations based on $E^T \cup E^L$ which are then compared with entries in E^P to finally measure the recommendation performance. In this way, recommendation methods with different number of free parameters can be evaluated and compared. Note that while introducing an additional parameter results in improved or at least unchanged performance in the usual train-probe approach, this is not necessarily the case in the threefold validation approach. From the machine learning point of view, threefold validation helps to avoid model over-fitting [29]. Table 2 demonstrates that the superiority of SPHY over HY is preserved also under threefold validation.

Table 2: Performance of the HY and SPHY method obtained under the threefold data division. The standard deviation of the results is again negligible with respect to the used numeric precision and the observed differences between HY and SPHY.

Data	Method	RS	$P(20)$	$H(20)$	$N(20)$
MovieLens	HY	0.081	0.113	0.88	179
	SPHY	0.078	0.120	0.90	168
Netflix	HY	0.054	0.103	0.61	996
	SPHY	0.048	0.113	0.75	862

We finally analyze computational complexity of the studied recommendation methods. Among them, MD and HC are parameter free and thus need no learning step. Since they are based on three steps of diffusion on a bipartite network, the computational complexity per user is $O(k_u^2 k_o)$ where k_u and k_o are, respectively, the mean degree of users and objects. SPMD, SPHC and HY have all one parameter each. Their computational complexity is $O((n+1)k_u^2 k_o)$, where n is the number of evaluation points of their respective parameters in the learning process. SPHY has two parameters, which results in computational complexity $O((n^2+1)k_u^2 k_o)$, The number of evaluations points n, however, is essentially independent of the data size and thus does not impairs scalability of methods that require the learning step (or steps).

Conclusion. – Spreading processes on networks have been extensively used in recommendation [12]. To increase the diversity of recommendations, we proposed two new processes: similarity-preferential mass diffusion (SPMD) and similarity-preferential heat conduction (SPHC). These processes contain a tunable parameter which makes it possible to gradually vary the preference given to similar users. Our results obtained on two standard data sets (MovieLens and Netflix) show that the hybrid of these two processes improves both diversity and accuracy of the produced recommendations. Our study of the optimal parameter values showed that they strongly depend on sparsity of the input data. When the data is relatively dense, a substantial increase of the weight of similar users produces the best results. With extremely sparse data, user weights are best to be made more equal by $\theta < 1$. We tested the new hybrid method also by threefold data division and showed that the observed improvement is not only an artifact caused by introducing an additional parameter but a real effect which can be used in practice.

The presented modification in essence resembles K-NN collaborative filtering methods which are based on considering only K most similar users (K nearest neighbors —hence the abbreviation) in the user-based collaborative filtering [30]. For suitably chosen K, these methods tend to perform better than unconstrained collaborative filtering where all users contribute proportionally to their similarity value. Note that by introducing the θ-parameter in the previously known MD, HC, and HY methods, we increased the weight of highly similar users at the expense of less similar users, thus effectively decreasing the number of users who contribute to the resulting recommendation scores. One can make the similarity closer by considering only K most similar users in MD, HC, and HY but our results show that the thus-achieved improvements are smaller than those of the presented similarity-preferential methods. Moreover, the generalization based on θ is very easy to implement and does not substantially increase the methods' computational complexity.

Our work can lead to many applications. For example, the link prediction in directed networks depends on a so-called "Bi-fan" structure [31]. The similarity preference can be introduced as a weighting strategy in this structure and improve the prediction precision. More generally, similarity-preferential diffusion can be applied to any multi-step diffusion process on networks. There are actually many different global and local diffusion based methods to estimate the similarity between nodes [20,32]. We believe that the similarity preference mechanism can improve them.

* * *

This work was partially supported by the Future and Emerging Technologies program of the European Commission FP7-COSI-ICT (project QLectives, grant no. 231200) and by the Swiss National Science Foundation (grant no. 200020-143272).

REFERENCES

[1] Resnick P. and Varian H. R., *Commun. ACM*, **40** (1997) 56.

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

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

[4] Balabanovic M. and Shoham Y., *Commun. ACM*, **40** (1997) 66.

[5] Koren Y., Bell R. and Volinsky C., *Computer*, **42** (2009) 30.

[6] Zhang Z.-K., Zhou T. and Zhang Y.-C., *Physica A*, **389** (2010) 179.

[7] Burke R., *Lect. Notes Comput. Sci.*, **4321** (2007) 377.

[8] Zeng W., Zeng A., Shang M.-S. and Zhang Y.-C., *Eur. Phys. J. B*, **86** (2013) 375.

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

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

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

[12] 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.

[13] Liu C. and Zhou W.-X., *Physica A*, **391** (2012) 5704.

[14] Zhou Y., Lü L., Liu W. and Zhang J., *PLoS ONE*, **8** (2013) e70094.

[15] Guan Y., Zhao D.-D., Zeng A. and Shang M.-S., *Physica A*, **392** (2013) 3417.

[16] Zhang F. G. and Zeng A., *EPL*, **100** (2012) 58005.

[17] Zhang Q.-M., Zeng A. and Shang M.-S., *PLoS ONE*, **8** (2013) e62624.

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

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

[20] Lü L., Medo M., Yeung C. H., Zhang Y.-C., Zhang Z.-K. and Zhou T., *Phys. Rep.*, **519** (2012) 1.

[21] Zhou T., Lü L. and Zhang Y. C., *Eur. Phys. J. B*, **71** (2009) 623.

[22] Grujić J., Mitrović M. and Tadić B., *Proceedings of the 16th International Conference on Digital Signal Processing (DSP)* (IEEE) 2009, p. 259.

[23] Yan B. and Gregory S., *Phys. Rev. E*, **85** (2012) 056112.

[24] Lancichinetti A. and Fortunato S., *Phys. Rev. E*, **84** (2011) 066122.

[25] Herlocker J. L., Konstan J. A., Borchers A. and Riedl J., *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM) 1999, p. 230.

[26] Bennett J. and Lanning S., *Proceedings of the KDD Cup Workshop* (ACM) 2007, p. 35.

[27] Lempel R. and Moran S., *Proceedings of the 12th International World Wide Web Conference* (ACM) 2003, p. 19.

[28] Ziegler C. N., McNee S. M., Konstan J. A. and Lausen G., *Proceedings of the 14th International World Wide Web Conference* (ACM) 2005, p. 22.

[29] Abu-Mostafa Y. S. and Magdon-Ismail M., *Learning From Data* (AMLBook) 2012.

[30] Sarwar B., Karypis G., Konstan J. and Riedl J., *Proceedings of the 10th International World Wide Web Conference* (ACM) 2001, p. 285.

[31] Zhang Q.-M., Lü L., Wang W.-Q., Zhu Y. X. and Zhou T., *PLoS ONE*, **8** (2013) e55437.

[32] Zeng A. and Lü L., *Phys. Rev. E*, **83** (2011) 056123.