Published in "Physica A 388(4): 462-468, 2009"
which should be cited to refer to this work.

Personal recommendation via modified collaborative filtering

Run-Ran Liu [a], Chun-Xiao Jia [a], Tao Zhou [a,b,*], Duo Sun [a], Bing-Hong Wang [a,c]

[a] *Department of Modern Physics and Nonlinear Science Center, University of Science and Technology of China, Hefei Anhui, 230026, PR China*
[b] *Department of Physics, University of Fribourg, Chemin du Muse 3, CH-1700 Fribourg, Switzerland*
[c] *Institute of Complex Adaptive System, Shanghai Academy of System Science, Shanghai, PR China*

PACS:
89.75.Hc
87.23.Ge
05.70.Ln

Keywords:
Recommendation system
Bipartite network
Similarity
Collaborative filtering
Infophysics

In this paper, we propose a novel method to compute the similarity between congeneric nodes in bipartite networks. Different from the standard cosine similarity, we take into account the influence of a node's degree. Substituting this new definition of similarity for the standard cosine similarity, we propose a modified collaborative filtering (MCF). Based on a benchmark database, we demonstrate the great improvement of algorithmic accuracy for both user-based MCF and object-based MCF.

1. Introduction

Recently, recommendation systems are attracting more and more attention, because it can help users to deal with information overload, which is a great challenge in modern society, especially under the exponential growth of the Internet [1] and the World-Wide-Web [2]. Recommendation algorithms have been used to recommend books and CDs at Amazon.com, movies at Netflix.com, and news at VERSIFI Technologies (formerly AdaptiveInfo.com) [3]. The simplest algorithm we can use in these systems is the global ranking method (GRM) [4], which sorts all the objects in descending order of degree and recommends those with the highest degrees. GRM is not a personal algorithm and its accuracy is not very high because it does not take personal preferences into account. Accordingly, various kinds of personal recommendation algorithms are proposed, for example, collaborative filtering (CF) [5,6], content-based methods [7,8], spectral analysis [9, 10], principal component analysis [11], the diffusion approach [4,12–14], and so on. However, the current generation of recommendation systems still requires further improvements to make recommendation methods more effective [3]. For example, content analysis is practical only if the items have well-defined attributes and those attributes can be extracted automatically; for some multimedia data, such as audio/video streams and graphical images, the content analysis is hard to apply. Collaborative filtering usually provides very bad predictions/recommendations to new users having very few collections. Spectral analysis has high computational complexity and is thus infeasible to deal with huge-size systems.

* Corresponding author at: Department of Modern Physics and Nonlinear Science Center, University of Science and Technology of China, Hefei Anhui, 230026, PR China.
E-mail address: zhutou@ustc.edu (T. Zhou).

Thus far, the widest applied personal recommendation algorithm is CF [3,15]. CF has two categories in general, one is user-based (U-CF), which recommends the target user the objects collected by the users sharing similar tastes; the other is object-based (O-CF), which recommends those objects similar to the ones the target user preferred in the past. In this paper, we introduce a modified collaborative filtering (MCF), which can be implemented for both object-based and user-based cases and achieve much higher accuracy of recommendation.

2. Method

We assume that there is a recommendation system which consists of m users and n objects, and each user has collected some objects. The relationship between users and objects can be described by a bipartite network. Bipartite network is a particular class of networks [4,16], whose nodes are divided into two sets, and connections among one set are not allowed. We use one set to represent users, and the other represents objects: if an object o_i is collected by a user u_j, there is an edge between o_i and u_j, and the corresponding element a_{ij} in the adjacent matrix A is set as 1, otherwise it is 0.

In U-CF, the predicted score v_{ij} (to what extent u_j likes o_i), is given as:

$$v_{ij} = \sum_{l=1,l\neq i}^{m} s_{il}a_{jl}, \tag{1}$$

where s_{il} denotes the similarity between u_i and u_l. For any user u_i, all v_{ij} are ranked by values from high to low, objects on the top and have not been collected by u_i are recommended.

How to determine the similarity between users? The most common approach taken in previous works focuses on the so-called structural equivalence. Two congeneric nodes (i.e. in the same set of a bipartite network) are considered structurally equivalent if they share many common neighbors. The number of common objects shared by users u_i and u_j is

$$c_{ij} = \sum_{l=1}^{n} a_{li}a_{lj}, \tag{2}$$

which can be regarded as a rudimentary measure of s_{il}. Generally, the similarity between u_i and u_j should be somewhat relative to their degrees [17]. There are at least three ways previously proposed to measure similarity, as:

$$s_{ij} = \frac{2c_{ij}}{k(u_i) + k(u_j)}, \tag{3}$$

$$s_{ij} = \frac{c_{ij}}{\sqrt{k(u_i)k(u_j)}}, \tag{4}$$

$$s_{ij} = \frac{c_{ij}}{\min(k(u_i), k(u_j))}. \tag{5}$$

The Eq. (3) is called Sorensen's index of similarity (SI) [18], which was proposed by Sorensen in 1948; the Eq. (4), called the cosine similarity, was proposed by Salton in 1983 and has a long history of the study on citation networks [17]. Both the Eqs. (4) and (5) are widely used in recommendation systems [3,4].

A common problem of Eqs. (3)–(5) is that they have not taken into account the influence of an object's degree, so that objects with different degrees have the same contribution to the similarity. If user u_i and u_j both have selected object o_l, that is to say, they have a similar taste for the object o_l. Provided that object o_l is very popular (the degree of o_l is very large), this taste (the favor for o_l) is a very ordinary taste and it does not mean u_i and u_j are very similar. Therefore, its contribution to s_{ij} should be small. On the other hand, provided that object o_l is very unpopular (the degree of o_l is very small), this taste is a peculiar taste, so its contribution to s_{ij} should be large. In other words, it is not very meaningful if two users both select a popular object, while if a very unpopular object is simultaneously selected by two users, there must be some common tastes shared by these two users. Accordingly, the contribution of object o_l to the similarity s_{ij} (if u_i and u_j both collected o_l) should be negatively correlated with its degree $k(o_l)$. We suppose the object o_l's contribution to s_{ij} being inversely proportional to $k^{\alpha}(o_l)$, with α a freely tunable parameter. The s_{ij}, consisted of all the contributions of commonly collected objects, is measured by the cosine similarity as shown in Eq. (4). Therefore, the proposed similarity reads:

$$s_{ij} = \frac{1}{\sqrt{k(u_i)k(u_j)}} \sum_{l=1}^{n} \frac{a_{li}a_{lj}}{k^{\alpha}(o_l)}. \tag{6}$$

Note that, the influence of an object's degree can also be embedded into the other two forms, shown in Eqs. (3) and (5), and the corresponding algorithmic accuracies will be improved too. Here in this paper, we only show the numerical results on cosine similarity as a typical example.

For any user-object pair u_i-o_j, if u_i has not yet collected o_j, the predicted score can be obtained by using Eq. (1). Here we do not normalize Eq. (1), because it will not affect the recommendation list, since for a given target user, we need sort all

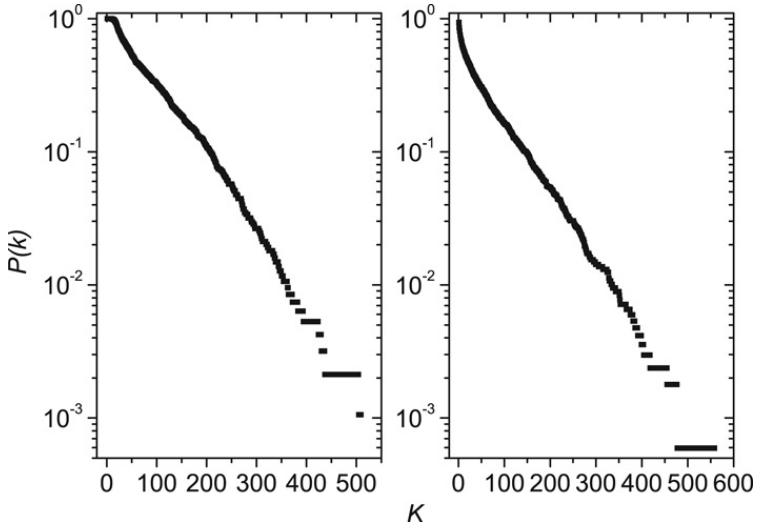

Fig. 1. The degree distributions of users (left panel) and objects (right panel) in linear-log plot, where $P(k)$ denotes the cumulative degree distribution.

her/his uncollected objects, and only the relative magnitude is meaningful. Note that, if two objects have exactly the same score, their order is randomly assigned. We call this method a modified user-based collaborative filtering (U-MCF), for it belongs to the framework of U-CF.

3. Numerical results

Using a benchmark data set namely *MovieLens* [19], we can evaluate the accuracy of the current algorithm. The data consists of 1682 movies (objects) and 943 users. Actually, *MovieLens* is a rating system, where each user votes movies in five discrete ratings 1–5. Hence we applied a coarse-grained method used in Refs. [4,12]: a movie has been collected by a user if and only if the rating given is at least 3 (i.e. the user at least likes this movie). The original data contains 10^5 ratings, 85.25% of which are $\geq$3, thus the data after coarse graining contains 85 250 user-object pairs. The current degree distributions of users and objects are presented in Fig. 1. Clearly, the degree distributions of both users and objects obey an exponential form. To test the recommendation algorithms, the data set is randomly divided into two parts: the training set contains 90% of the data, and the remaining 10% of data constitutes the probe. Of course, we could divide it in other proportions, for example, 80% *vs*. 20%, 70% *vs*. 30%, and so on. The training set is treated as known information, while no information in the probe set is allowed to be used for prediction.

A recommendation algorithm could provide each user a recommendation list which contains all her/his uncollected objects. There are several measures for evaluating the quality of these recommendation lists generated by different algorithms. In this paper, we use *ranking score, recall* and *precision* to measure the effectiveness of a given recommendation approach. A good overview of these measures can be found in Ref. [6].

Ranking score. For an arbitrary user u_i, if the relation u_i-o_j is in the probe set (according to the training set, o_j is an uncollected object for u_i), we measure the position of o_j in the ordered list. For example, if there are 1000 uncollected movies for u_i, and o_j is the 10th from the top, we say the position of o_j is the top 10/1000, denoted by $r_{ij} = 0.01$. Since the probe entries are actually collected by users, a good algorithm is expected to give high recommendations to them, thus leading to small r. Therefore, the mean value of the position value $\langle r \rangle$ (called ranking score [4]), averaged over all the entries in the probe, can be used to evaluate the algorithmic accuracy. The smaller the ranking score, the higher the algorithmic accuracy, and vice versa. The definition of ranking score here is slightly different from that of the Ref. [4]. It is because if a movie or user in the probe set has not yet appeared in the training set, we automatically remove it from the probe and the number of total movies was counted only for the ones appeared in the the training set; while the Ref. [4] takes into account those movies only appeared in the probe via assigning zero score to them. This slight difference in implementation does not affect the conclusion.

Recall is defined as the ratio of number of recommended objects appeared in the probe to the total number of data entries in the probe. The larger recall corresponds to the better performance.

Precision is defined as the ratio of number of recommended objects appeared in the probe to the total number of recommended objects. A larger precision corresponds to a better performance. Precision is also called the hitting rate in the literature [4].

Recall and precision can be used to realize the balance of two competitive factors: cost and efficiency. The cost denotes the total number of recommended movies, while the efficiency denotes the total number of movies that are recommended correctly. The efficiency can be improved by increasing the number of recommended movies; however, the cost is increasing at the same time. That is to say, the cost can be decreased by reducing the recommendations, while the efficiency may be

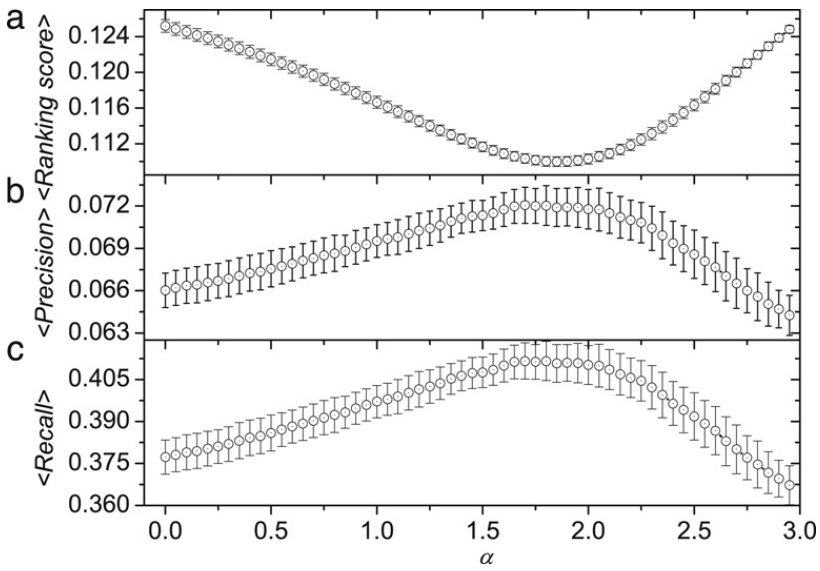

Fig. 2. The effect of parameter α in U-MCF. The ranking score has its minimum at about $\alpha = 1.85$, at almost the same point, the recall and precision achieve their maximums. Present results are obtained by averaging over four independent 90% *vs.* 10% divisions. The error bars denote the standard deviations.

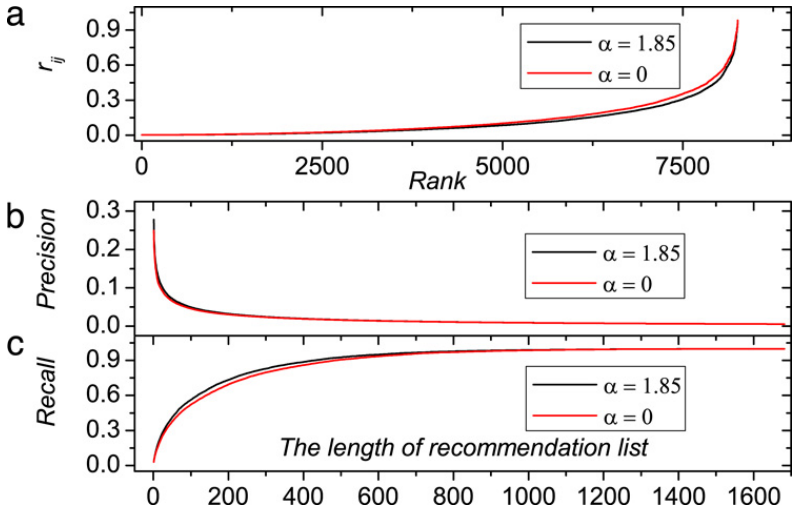

Fig. 3. (Color online) (a) The predicted position of each entry in the probe ranked in ascending order. (b) The precision for different lengths of recommendation lists. (c) The recall for different lengths of recommendation lists.

decreased correspondingly. Consequently, we use precision and recall balancing these two competitive factors. At a certain length of recommendation list L, precision tests whether the cost is deserved or necessary, while recall tests whether the efficiency is sufficient. Based on these two measures, one can find a certain L as a tradeoff for cost and efficiency.

Fig. 2 reports the algorithmic accuracy of U-MCF, which has a clear optimal case around $\alpha = 1.85$. Fig. 3(a) reports the distribution of all the position values, r_{ij}, which are sorted from the top position ($r_{ij} \to 0$) to the bottom position ($r_{ij} \to 1$). Fig. 3(b) and (c) report the recall and precision for different lengths of recommendation lists respectively. We set L as 50 in our numerical experiment (in real e-commerce systems, the length of recommendation list usually ranges from 10 to 100 [20]), therefore the total number of recommended objects is $mL = 47150$. Fig. 4 reports the algorithmic accuracies of the standard case ($\alpha = 0$) and the the optimal cases ($\alpha = 1.85$) for different sizes of training sets. All these numerical results strongly demonstrate that to depress the contribution of common selected popular objects can further improve the algorithmic accuracy.

Similar to the U-CF, the recommendation list can also be obtained by object-based collaborative filtering (O-CF), that is to say, the user will be recommended objects similar to the ones he/she preferred in the past [21]. By using the cosine expression, the similarity between two objects, o_i and o_j, can be written as:

$$s_{ij} = \frac{1}{\sqrt{k(o_i)k(o_j)}} \sum_{l=1}^{m} a_{il}a_{jl}. \tag{7}$$

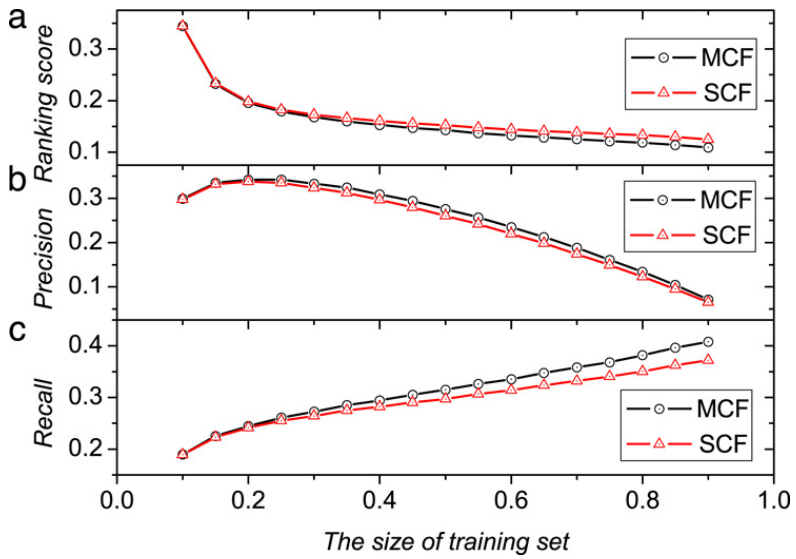

Fig. 4. (Color online) The standard CF (SCF) (i.e. $\alpha = 0$) *vs*. the optimal case for different sizes of training sets.

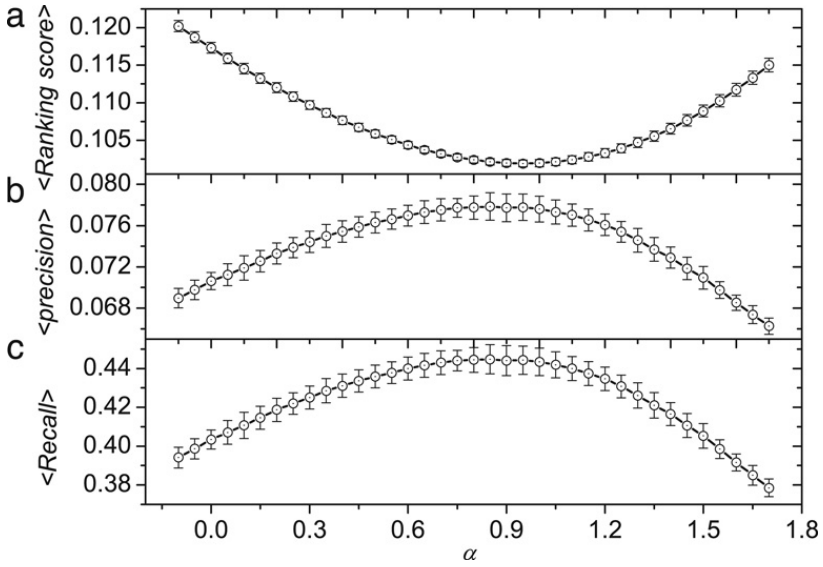

Fig. 5. The effect of parameter α in O-MCF. The ranking score has its minimum at about $\alpha = 0.95$, at almost the same point, the recall and precision achieve their maximums. Present results are obtained by averaging over four independent 90% *vs*. 10% divisions. The error bars denote the standard deviations.

The predicted score, to what extent u_i likes o_j, is given as:

$$v_{ij} = \sum_{l=1, l\neq i}^{n} s_{jl} a_{li}. \tag{8}$$

Analogously, taking into account the influence of user degree, a modified expression of object–object similarity reads:

$$s_{ij} = \frac{1}{\sqrt{k(o_i)k(o_j)}} \sum_{l=1}^{m} \frac{a_{il}a_{jl}}{k^{\alpha}(u_l)}, \tag{9}$$

where α is a free parameter. The modified object-based collaborative filtering (O-MCF for short) can be obtained by combining Eqs. (8) and (9). Fig. 5 reports the algorithmic accuracy of O-MCF, which has a clear optimal case around $\alpha = 0.95$. Fig. 6(a) reports the distribution of all the position values, r_{ij}, which are sorted from the top position ($r_{ij} \to 0$) to the bottom position ($r_{ij} \to 1$), Fig. 6(b) and (c) report the recall and precision for different lengths of recommendation lists respectively. Fig. 7 reports the algorithmic accuracies of the standard case ($\alpha = 0$) and the the optimal case ($\alpha = 0.95$) for different sizes of training sets. All these results, again, demonstrate that to depress the contribution of users with high degrees of object–object similarity can further improve the algorithmic accuracy of object-based method.

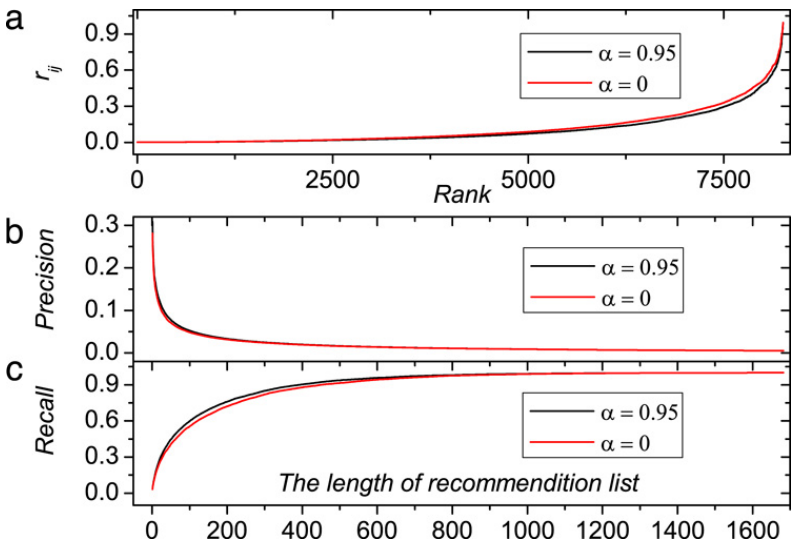

Fig. 6. (Color online) Similar to Fig. 3. But for O-MCF.

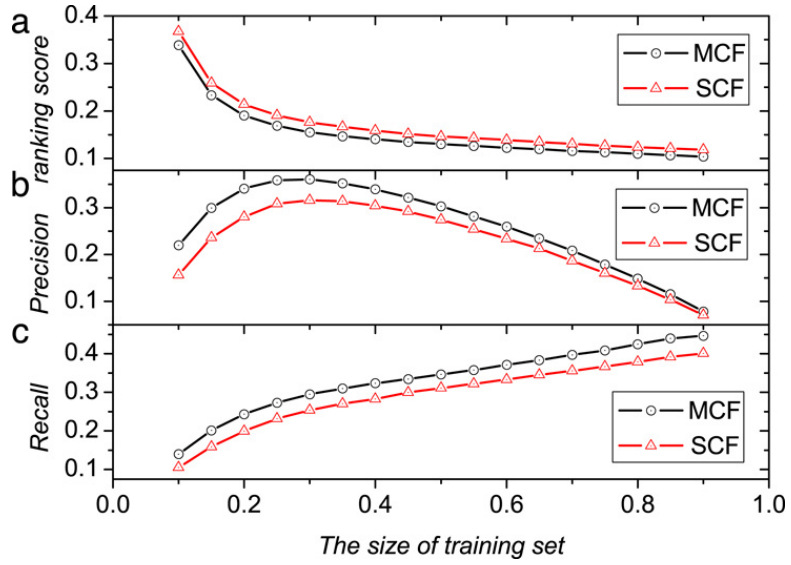

Fig. 7. (Color online) Similar to Fig. 4. But for O-MCF.

Table 1
Three measures for different algorithms with a probe set containing 10% data. For precision and recall, $L = 50$. Present results are obtained by averaging over four independent divisions. The values corresponding to U-MCF and O-MCF are the optimal ones.

Method	⟨Ranking score⟩	⟨Recall⟩	⟨Precision⟩
GRM	0.1502	0.3077	0.0540
O-CF	0.1173	0.4035	0.0706
U-CF	0.1252	0.3773	0.0660
O-MCF	0.1019	0.4443	0.0777
U-MCF	0.1101	0.4108	0.0719

4. Conclusion

We compare the MCF, standard CF and GRM in Table 1. Clearly, MCF is the best method and GRM performs worst. Compared with the standard CF, the modified object-based algorithm and the modified user-based method improve the accuracy to a different extent in three measures. Ignoring the degree-degree correlation in user-object relations, the algorithmic complexity of U-MCF is $O(m^2\langle k_u\rangle + mn\langle k_o\rangle)$, the O-MCF is $O(n^2\langle k_o\rangle + mn\langle k_u\rangle)$, respectively. Here $\langle k_u\rangle$ and $\langle k_o\rangle$ denote the average degree of users and objects. Therefore, one can choose either O-MCF or U-MCF according to specific properties of the data source. For example, if the user number is much larger than the object number (i.e. $m \gg n$), the O-MCF runs much faster. On the contrary, if $n \gg m$, the U-MCF runs faster. Furthermore, the remarkable improvement of algorithmic accuracy also indicates that our definition of similarity is more reasonable than the traditional one.

Acknowledgments

The authors would like to acknowledge Dr. Jianguo Liu for valuable discussion and *GroupLens Research Group* for providing us the data set *MovieLens*. This work is funded by the National Basic Research Program of China (973 Program No.2006CB705500), the National Natural Science Foundation of China (Grant Nos. 60744003, 10635040 and 10532060). T.Z. acknowledges the support from SBF (Switzerland) for financial support through project C05.0148 (Physics of Risk), and the Swiss National Science Foundation (205120-113842).

References

[1] M. Faloutsos, P. Faloutsos, C. Faloutsos, Comput. Commun. Rev. 29 (1999) 251.
[2] A. Broder, R. Kumar, F. Moghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Comput. Netw. 33 (2000) 309.
[3] G. Adomavicius, A. Tuzhilin, IEEE Trans. Knowl. Data Eng. 17 (2005) 734.
[4] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, Phys. Rev. E 76 (2007) 046115.
[5] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, Commun. ACM 40 (1997) 77.
[6] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, ACM Trans. Inf. Syst. 22 (2004) 5.
[7] M. Balabanovi, Y. Shoham, Commun. ACM 40 (1997) 66.
[8] M.J. Pazzani, Artif. Intell. Rev. 13 (1999) 393.
[9] S. Maslov, Y.-C. Zhang, Phys. Rev. Lett. 87 (2001) 248701.
[10] J. Ren, T. Zhou, Y.-C. Zhang, Europhys. Lett. 82 (2008) 58007.
[11] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Inform. Ret. 4 (2001) 133.
[12] T. Zhou, L.-L. Jiang, R.-Q. Su, Y.-C. Zhang, Europhys. Lett. 81 (2008) 58004.
[13] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, F. Yang, Europhys. Lett. 80 (2007) 68003.
[14] J.-G. Liu, T. Zhou, B.-H. Wang, Y.-C. Zhang, arXiv: 0808.3726.
[15] Z. Huang, H. Chen, D. Zeng, ACM Trans. Inf. Syst. 22 (2004) 116.
[16] P. Holme, F. Liljeros, C.R. Edling, B.J. Kim, Phys. Rev. E 68 (2003) 056107.
[17] E.A. Leicht, P. Holme, M.E.J. Newman, Phys. Rev. E 73 (2006) 026120.
[18] T. Sorenson, Biol. Skr. 5 (1948) 1.
[19] The MovieLens data can be downloaded from the website of GroupLens Research. http://www.grouplens.org.
[20] J.B. Schafer, J.A. Konstan, J. Riedl, Data Min. Knowl. Discov. 5 (2001) 115.
[21] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Proc. 10th Int. Conf. WWW, 2001, pp. 285–295.