Solving the accuracy-diversity dilemma via directed random walks

Jian-Guo Liu,[1,2,*] Kerui Shi,[1] and Qiang Guo[1]

[1]Research Center of Complex Systems Science, University of Shanghai for Science and Technology, Shanghai 200093, People's Republic of China

[2]CABDyN Complexity Center, Saïd Business School, University of Oxford, Park End Street, Oxford, OX1 1HP, United Kingdom

(Received 18 October 2011; revised manuscript received 24 December 2011; published 30 January 2012)

Random walks have been successfully used to measure user or object similarities in collaborative filtering (CF) recommender systems, which is of high accuracy but low diversity. A key challenge of a CF system is that the reliably accurate results are obtained with the help of peers' recommendation, but the most useful individual recommendations are hard to be found among diverse niche objects. In this paper we investigate the direction effect of the random walk on user similarity measurements and find that the user similarity, calculated by directed random walks, is reverse to the initial node's degree. Since the ratio of small-degree users to large-degree users is very large in real data sets, the large-degree users' selections are recommended extensively by traditional CF algorithms. By tuning the user similarity direction from neighbors to the target user, we introduce a new algorithm specifically to address the challenge of diversity of CF and show how it can be used to solve the accuracy-diversity dilemma. Without relying on any context-specific information, we are able to obtain accurate and diverse recommendations, which outperforms the state-of-the-art CF methods. This work suggests that the random-walk direction is an important factor to improve the personalized recommendation performance.

DOI: 10.1103/PhysRevE.85.016118 PACS number(s): 89.75.Hc, 89.20.Hh, 05.70.Ln

I. INTRODUCTION

Due to the rapidly expanding internet and social network, we are overloaded by the unlimited information on the World Wide Web [1]. For instance, one has to choose among thousands of candidate commodities to shop online and finds the relevant information from billions of Web pages. Comprehensive exploration for each user is infeasible [2]. Consequently, how to efficiently help people obtain information that they truly need is a challenging task nowadays [3]. A landmark for information filtering is the use of a search engine, by which users could find the relevant Web pages with the help of properly chosen keywords [4,5]. However, sometimes users' tastes or preferences evolve with time and cannot be accurately expressed by keywords, and search engines do not take into account the personalization and tend to return the same results for people with far different needs. Being an effective tool to address this problem, recommender systems have become a promising way to filter out the irrelevant information and recommend potentially interesting items to the target user by analyzing their interests and habits through their historical behaviors [3,6–12]. Motivated by its significance in economy and society, the design of an efficient recommendation algorithm has become a joint focus of theoretical physics [13,14], computer science [3], and management science [8].

Zhang *et al.* [13] proposed a new information framework based on the heat conduction process, namely, the heat-conduction-based (HC) recommendation model. The HC model supposes that the objects one user has collected have the recommendation power to help the target user find potentially relevant objects. If the target user is replaced by a specific object, the HC model is similar to the collaborative filtering (CF) method, in which the users-rated target objects have the recommendation powers to identify the potentially interesting users. So far, the CF method has been successfully applied to many online applications and has become one of the most successful technologies for recommender systems [9–12,15,16]. For example, Herlocker *et al.* [17] proposed an algorithmic framework referring to the user similarity. Luo *et al.* [18] introduced the concepts of local and global user similarity based on surprisal-based vector similarity and the concept of maximum distance in graph theory. Sarwar *et al.* [19] proposed the item-based CF algorithm by comparing different items. Deshpande and Karypis [20] proposed the item-based top-N CF algorithm, in which items are ranked according to the frequency of appearing in the set of similar items and the top-N ranked items are returned. Gao *et al.* [21] incorporated the user ranking information into the computation of item similarity to improve the performance of the item-based CF algorithm. Recently some physical dynamics, including random walks [10,22,23] and the heat conduction process [13], has found applications in node similarity measurement. Liu *et al.* [10] used the random walks to calculate the user similarity and found that the modified CF algorithm has remarkably higher accuracy. As a benchmark for comparison, we call it the standard CF algorithm (hereinafter CF stands only for the collaborative filtering using random-walk-based user similarity [10]). By considering the high-order correlation of the users and objects, Zhou *et al.* [9] and Liu *et al.* [11] proposed the ultra-accurate algorithms, in which the second-order correlations are used to delete the redundant information. Besides reliably accurate recommendations, it is also important for recommender systems to help most individuals find diverse niche objects. CF algorithms generate recommendations according to similar users' suggestions, which prefers ranking the popular objects at the top positions of recommendation lists, leading to high accuracy but low diversity.

Random walks have been used to quantify trajectories in a symmetric ways, namely, in- and out-diversity and accessibility [24–26]. In this paper we argue that the opinions

*jianguo.liu@sbs.ox.ac.uk

of small-degree users should be enhanced to generate diverse recommendations and present a new directed random-walk-based CF algorithm, namely the NCF algorithm, to investigate the effect of user similarity direction on recommendations. The numerical results on the data sets, *Netflix* and *MovieLens*, show that the accuracy of NCF outperforms the state-of-the-art CF methods with greatly improved diversity, which suggests that the similarity direction is an important factor for user-based information filtering.

II. BIPARTITE NETWORK AND HEAT-CONDUCTION-BASED MODEL

A recommendation system consists of a set of nodes, object nodes, and connections between two nodes corresponding to an object voted on or collected by a user, which could be represented by a bipartite network $G(U,O,E)$. We denote the object set as $O = \{o_1,o_2,\ldots,o_m\}$, the user set as $U = \{u_1,u_2,\ldots,u_n\}$, and the connection set as $E = \{e_1,e_2,\ldots,e_q\}$. The bipartite network can then be represented by an adjacent matrix $A = \{a_{\alpha j}\} \in R^{m,n}$, where $a_{\alpha j} = 1$ if o_α is collected by u_j, and $a_{\alpha j} = 0$ otherwise.

The final aim of recommender systems is to identify a given user's interesting objects and generate a ranking list of the target user's uncollected objects according to the predicted scores. The HC model supposes the neighbor nodes of one target node as the heat sources with temperature 1, while the remaining nodes are of temperature 0. According to the thermal equilibrium [13], the temperature of the remaining nodes is set as the predicted scores, which could be calculated by solving the equation $\mathbf{W}^h H = \mathbf{f}$, where $\mathbf{f}$ is the flux vector [13]. The standard HC model first constructs the propagator matrix $\mathbf{W}^h$, where the element $w_{\alpha\beta}$ denotes the conduction rate from object o_β to o_α, and sets the temperatures of the target node's neighbors as 1; then the heat will diffuse between heat sources and other nodes. Finally, the temperatures of uncollected objects are considered as recommendation scores.

The general framework of the item-based HC model is as follows: (1) construct the weighted object network (i.e., determine the matrix $\mathbf{W}$) from the known user-object relations; (2) determine the initial resource vector $\mathbf{f}$ for each user; (3) get the final resource distribution via

$$\mathbf{f}' = \mathbf{W}\mathbf{f}; \tag{1}$$

and (4) recommend those uncollected objects with highest final scores. Note that the initial configuration $\mathbf{f}$ is determined by the user's personal information; thus for different users, the initial configuration is different. For a given object o_α, the ith element of $\mathbf{f}^\alpha$ should be zero if $a_{\alpha i} = 0$. That is to say, one should not put any recommendation power (i.e., resource) onto an unrated user. The simplest case is to set a uniformly initial configuration as

$$f_i^\alpha = a_{\alpha i}. \tag{2}$$

Under this configuration, all the users that rated object o_α have the same recommendation power.

The traditional HC model is implemented based on the object similarity. If the number of users is much smaller than the one of objects, we could apply a similar idea based on the user similarity, namely, the user-based HC model. From the definition, one can find that the user-based HC model is equivalent to the CF algorithm; therefore, the user similarity analysis of the CF algorithm could also bring deep insight into the HC model. Zhou *et al.* [23] used the random-walk process to calculate the node similarity of bipartite networks and proposed the network-based inference (NBI) recommendation algorithm [27]. In Ref. [14], the NBI algorithm was also referred to as the Probs algorithm. Liu *et al.* [10] embedded the random-walk process into the CF algorithm to calculate the user similarity and found that the algorithmic accuracy is greatly increased. In Ref. [10] a certain amount of resource is associated with each user, and the weight s_{ij} represents the proportion of the resource u_j would like to distribute to u_i. This process works by using the random-walk process on user-object bipartite networks, where each user distributes his or her initial resource equally to all the objects he or she has collected, and then each object sends back what it has received to all the users who collect it. The weight s_{ij} representing the amount of initial resource u_j evenly transferred to u_i can be defined as

$$s_{ij} = \frac{1}{k_{u_j}} \sum_{l=1}^{m} \frac{a_{li} a_{lj}}{k_{o_l}}, \tag{3}$$

where k_{u_i} and k_{o_l} indicate the degrees of user u_i and object o_l.

In the random-walk process, the user similarity from user u_j to u_i, s_{ij}, is determined by the degrees of commonly rated objects and user u_j's degree k_{u_j}. It is unlikely these quantities are exactly the same for each pair of users, and therefore, $s_{ij} \neq s_{ji}$ in most cases.

III. EFFECT OF USER SIMILARITY DIRECTION ON CF ALGORITHM

According to the random-walk-based user similarity calculation (see Fig. 1), the user similarity is calculated by random walks in a symmetric way. In the CF algorithm, the system

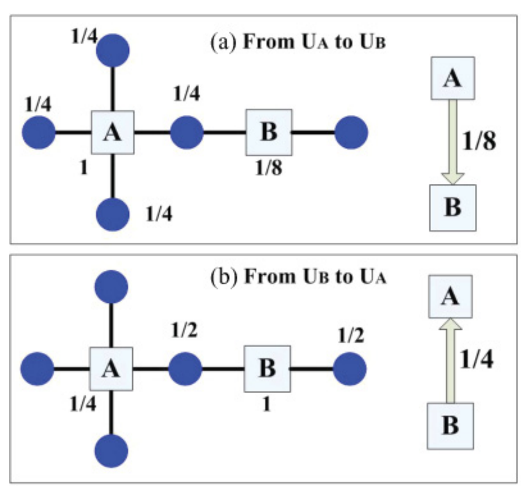

FIG. 1. (Color online) Illustration of random-walk-based user similarity calculation, which has been used to measure user or object similarity in personalized recommendations. (a) The possibility walking from user u_A to user u_B is used to measure the directed user similarity $s_{BA} = 1/8$. (b) Similarity between user u_B to user u_A is $s_{AB} = 1/4$. The degrees of user u_A and u_B are $k_{u_A} = 4$ and $k_{u_B} = 2$, and one has $\frac{k_{u_A}}{k_{u_B}} = \frac{0.25}{0.125}$.

should identify the target user's interesting objects with the help of his neighbors' historical selections or collections. Therefore, after we obtain the user similarity matrix, the similarities between neighbors to the target user are used to evaluate the predicted scores. According to Eq. (3), for one pair of users u_i and u_j, their similarities could be written as

$$s_{ij} = \frac{1}{k_{u_j}} \sum_{l=1}^{m} \frac{a_{li}a_{lj}}{k_{o_l}}, \quad \text{from } u_j \text{ to } u_i,$$
$$s_{ji} = \frac{1}{k_{u_i}} \sum_{l=1}^{m} \frac{a_{li}a_{lj}}{k_{o_l}}, \quad \text{from } u_i \text{ to } u_j. \tag{4}$$

Therefore, one has

$$\frac{s_{ij}}{s_{ji}} = \frac{k_{u_i}}{k_{u_j}}. \tag{5}$$

If $k_{u_i} > k_{u_j}$, then $s_{ij} > s_{ji}$ and vice versa. For MovieLens and Netflix data sets, the exponential forms of user degree distributions indicate that most users' degree are very small (see Fig. 2), which means that large-degree users would frequently be identified as small-degree users' friends. As a consequence, most users' recommendation lists would be similar.

In order to investigate the effect of user similarity direction on CF algorithms, we introduce a new user similarity direction generated by random walks from neighbor set U_n to the target user to measure user similarities and calculate the predicted score $v_{i\alpha}$. The NCF algorithm could be described as follows: First, calculate directed user similarities according to Eq. (3); then calculate the predicted scores for target user u_i's uncollected objects by

$$v_{i\alpha} = \frac{\sum_{j=1}^{n} s_{ij}^{\beta} a_{\alpha j}}{\sum_{ij}^{n} s_{ij}^{\beta}}, \tag{6}$$

where β is a tunable parameter to investigate the effect of similarity strength on the algorithmic performance, and s_{ij} is the similarity from user u_j to u_i. When $\beta = 1$, all the user similarities are given the same weight; when $\beta > 1$, the preferences of users with larger similarities are strengthened; when $\beta < 1$, the ones with smaller similarities are strengthened. The numerical results indicate that changing the user similarity direction could not only accurately identify user's interests, but also increase the algorithmic capability of finding niche objects.

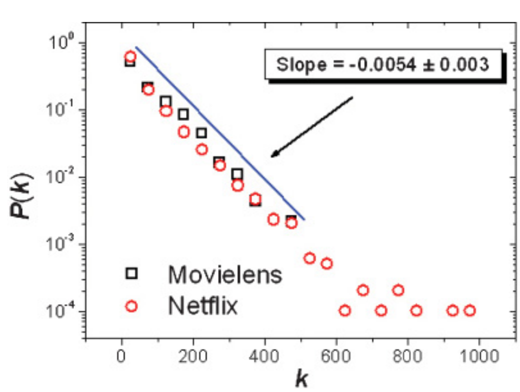

FIG. 2. (Color online) User degree distributions for MovieLens and Netflix data sets, which approximately have exponential forms $P(k) \sim \exp(-0.0054k \pm 0.003)$.

IV. MAXIMAL-SIMILARITY-BASED CF ALGORITHM

The algorithmic performance may be affected by the user similarity direction and may also be determined by the properties of the data set. In other words, although the algorithmic performance of the NCF algorithm is much better than the one of the CF algorithm, it may happen only on specific data sets whose similarities between neighbors and the target user are more effective than the ones in the opposite direction. In order to make it clear, we present a maximal-similarity-based CF (MCF) algorithm to investigate the influence of similarity magnitude, in which the predicted score from user u_i to the uncollected object o_α, $v_{i\alpha}$, is given by

$$v_{i\alpha} = \frac{\sum_{j=1}^{n} s_{\max}^{\beta} a_{\alpha j}}{\sum_{j=1}^{n} s_{\max}^{\beta}}, \tag{7}$$

where $s_{\max}$ is defined as the larger similarity between user u_i and u_j:

$$s_{\max} = \max\{s_{ij}, s_{ji}\}. \tag{8}$$

For example, the similarity from u_i to u_j is $s_{ji} = 0.01$, while the one from u_j to u_i is $s_{ij} = 0.9$. When recommending objects to u_i, the larger similarity 0.9 is used regardless of the similarity direction.

V. SIMULATION RESULTS

A. Data description

In this paper we base our simulation results on two data sets. The MovieLens[1] data set consists of 100 000 ratings from 943 users on 1574 movies (objects) and rating scale from one (i.e., worst) to five (i.e., best). The Netflix data set [28] is a random sample of the whole records of user activities on Netflix.com, which consists of 6000 movies, 10 000 users, and 824 802 ratings. The users of Netflix also vote on movies with discrete ratings from one to five. Here we apply a coarse graining method: A movie is considered to be collected by a user only if the rating is larger than two. In this way, the MovieLens data have 82 580 edges, and the Netflix data have 701 947 edges (see Table I for basic statistics). As an online movie recommendation Web site, MovieLens invites users to rate movies and, in return, makes personalized recommendations and predictions for movies the user has not already rated; undercontribution is common. Unlike the Netflix web site, MovieLens does not have any DVD rental service. The data set E is randomly divided into two parts, $E = E^T \cup E^P$, where the training set E^T is treated as known information, containing p percent of the data, and the remaining $1 - p$ part is set as the probe set E^P, whose information is not allowed to be used for prediction.

[1] http://www.MovieLens.org.

TABLE I. Basic statistics of the tested data sets.

Data sets	Users	Objects	Links	Sparsity
MovieLens	1574	943	82 580	5.56×10^{-2}
Netflix	10 000	6000	701 947	1.17×10^{-2}

B. Metrics

1. Average ranking score

Accuracy is one of the most important metrics to evaluate the recommendation algorithmic performance. An accurate method will put preferable objects in higher places. Here we use the *average ranking score* [23] to measure the accuracy of the algorithm. For an arbitrary user u_i, if the object o_α is not collected by user u_i, while the entry u_i-o_α is in the probe set, we use the rank of o_α in the recommendation list to evaluate accuracy. For example, if there are eight uncollected objects for user u_i, and object o_α is ordered at the third place, we say the position of o_α is 3/8, denoted by $r_{i\alpha} = 0.375$. Since the probe entries are actually collected by users, a good algorithm is expected to give high recommendations to them, leading to a small $r_{i\alpha}$. Therefore, the mean value of the positions, averaged over all the entries in the probe set, can be used to evaluate the algorithmic accuracy:

$$\langle r \rangle = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\sum_{(u_i,o_\alpha)\in E^p} r_{i\alpha}}{\theta - k_{u_i}}\right), \tag{9}$$

where E^p is the edge set existing in the probe set and θ is the number of objects in the probe set. The smaller the average ranking score, the higher the algorithmic accuracy, and vice versa.

2. Precision and recall

Since users usually consider only the top part of the recommendation list, a more practical metric is to consider the number of users' hidden links ranked in the top-L places. We adopt another accuracy measure called *precision*. For a target user, the precision is defined as the ratio between relevant objects (namely, the objects collected by u_i in the probe set) and the length L. Averaging the individual precisions over all users, we obtain the mean value $P(L)$ of the algorithm on one data set,

$$P(L) = \frac{1}{n}\sum_{i}\frac{d_i(L)}{L}, \tag{10}$$

where $d_i(L)$ indicates the number of relevant objects existing in the top-L places of the recommendation list. A larger precision corresponds to a better performance. *Recall* is defined as the ratio between the number of objects existing in the top-L places of the recommendation list and the total number of collected objects C_i in the probe set. Averaging over the individual recalls, we obtain the mean recall $R(L)$, which could be defined as

$$R(L) = \frac{1}{n}\sum_{i}\frac{d_i(L)}{C_i}. \tag{11}$$

The larger recall corresponds to the better performance.

3. Diversity

The analysis results on the Facebook data set showed that, besides the common interests, users of online social networks also have their specific tastes and interests [29], leading to diverse selection behaviors. Liu *et al.* [30] found that users' tastes reflected in MovieLens and Netflix data could also be divided into two categories: *common interests* and *specific interests*. Therefore, besides accuracy, the diversity of all recommendation lists is taken into account to evaluate the algorithmic performance. In general, most of the users would not show a negative altitude to popular objects; therefore, ranking popular objects at the top part of recommendation lists would generate higher accuracy. However, personalized recommendation algorithms should not only present accurate prediction but also generate different recommendations to different users according to their specific tastes or habits. The diversity can be quantified by the average Hamming distance:

$$S = 1 - \langle Q_{ij}(L)\rangle/L, \tag{12}$$

where L is the length of the recommendation list and Q_{ij} is the number of overlapped objects in u_i's and u_j's recommendation lists. The largest $S = 1$ indicates recommendations to all users are completely different; in other words, the system has the highest diversity, while the smallest $S = 0$ means all of the recommendations are exactly the same.

4. Popularity

An accurate and diverse recommender system is expected to help users find the niche or unpopular objects that are hard for them to identify. The metric *popularity* is introduced to quantify the capacity of an algorithm to generate unexpected recommendation lists, which are defined as the average collected times over all recommended objects:

$$\langle k \rangle = \frac{1}{n}\sum_{i}\left(\frac{1}{L}\sum_{o_\alpha\in O_i^L} k_{o_\alpha}\right), \tag{13}$$

where O_i^L is u_i's recommendation list with length L. A smaller average degree $\langle k\rangle$, corresponding to less popular objects, is preferred since those lower-degree objects are hard to be found by users themselves.

C. Simulation results

We summarize the results for NCF, CF, and MCF algorithms, as well as the metrics for MovieLens and Netflix data sets, in Table II. Clearly, NCF outperforms the classical CF and MCF algorithms over all five metrics, including average ranking score $\langle r\rangle$, diversity S, popularity $\langle k\rangle$, precision P, and recall R. Table III gives the comparisons among different algorithms for $p = 0.9$. The so-called optimal parameters are subject to the lowest average ranking score $\langle r\rangle$. The metrics, including average ranking score $\langle r\rangle$, diversity S, and popularity $\langle k\rangle$, are obtained at the optimal parameters. From which one can see that the accuracy of NCF is close to the result of hybrid algorithm [14] and outperforms the state-of-the-art CF algorithms using the second-order correlation information [11]. Among all algorithms, the Heter-CF algorithm gives the highest diversity, while the CB-CF algorithm generates the lowest popularity. Comparing with these two outstanding

TABLE II. Performances of NCF, CF, and MCF algorithms for Netflix and MovieLens data sets according to each of five metrics. The popularity $\langle k \rangle$, diversity S, precision P, and recall R corresponding to $L = 10$.

		$\langle r \rangle$	$\langle k \rangle$	S	P	R
Netflix	NCF	**0.0450**	**2506**	**0.8236**	**0.0967**	**0.1640**
	CF	0.0497	2813	0.7001	0.0917	0.1365
	MCF	0.0477	2758	0.7378	0.0954	0.1374
MovieLens	NCF	**0.0864**	**237**	**0.8929**	**0.1502**	**0.2037**
	CF	0.1037	275	0.8435	0.1497	0.2010
	MCF	0.0970	271	0.8434	0.1459	0.1936

algorithms, NCF can reach or closely approach the best diversity without taking into account high-order correlation and provide more accurate recommendation results.

Figure 3 reports the algorithmic accuracy as a function of β. In our algorithm the curve (the red circle) has a clear minimum around $\beta_{\rm opt} = 3.3$ for MovieLens and $\beta_{\rm opt} = 2.0$ for Netflix. Comparing with CF algorithms whose user similarities are defined from large-degree to small-degree users, the average ranking score $\langle r \rangle$ of NCF is reduced from 0.497 to 0.045 for Netflix and from 0.1037 to 0.0864 for the MovieLens data set, reduced by 9.9% and 16.68%, respectively, at the optimal values. Comparing with the MCF algorithm, the performance of NCF is also better. Subject to the accuracy, the reason why NCF outperforms CF and MCF indeed lies in the direction effect but not the data effect, and the results also indicate that giving more recommendation power to the small-degree users could enhance the accuracy and diversity simultaneously.

The Hamming distance S is introduced to measure the algorithmic performance to present personalized recommendation lists. The average object degree $\langle k \rangle$ is used to evaluate the ability that an algorithm gives a novel recommendation. Figures 2(a)–2(d) shows $\langle k \rangle$ and S as a function of β when recommendation list length $L = 10$, respectively. For the MovieLens data set, at the optimal point $\beta_{\rm opt} = 3.3$, the

TABLE III. Algorithmic performances for MovieLens data when $p = 0.9$, including the average ranking score $\langle r \rangle$, diversity S, and popularity $\langle k \rangle$ corresponding to length of recommendation list $L = 50$. GRM is a global ranking method; CF is the collaborative filtering algorithm based on random walks [10]; Heter-CF is a modified CF algorithm, in which the user similarity is defined based on the mass diffusion process, and the second-order similarity is involved ($\beta_{\rm opt} = -0.82$) [11]. CB-CF refers to the CF algorithm on weighted bipartite network [15]; Hybrid is an abbreviation for the hybrid algorithm proposed in Ref. [14]. Each number is obtained by a averaging over ten runs of independently random division of training set and probe set.

Algorithms	Ranking score	Popularity	Hamming distance
GRM	0.1390	259	0.398
CF	0.1063	229	0.750
Heter-CF	0.0877	175	0.826
CB-CF	0.0914	148	0.763
Hybrid	0.0850	167	0.821
NCF	0.0864	178	0.801

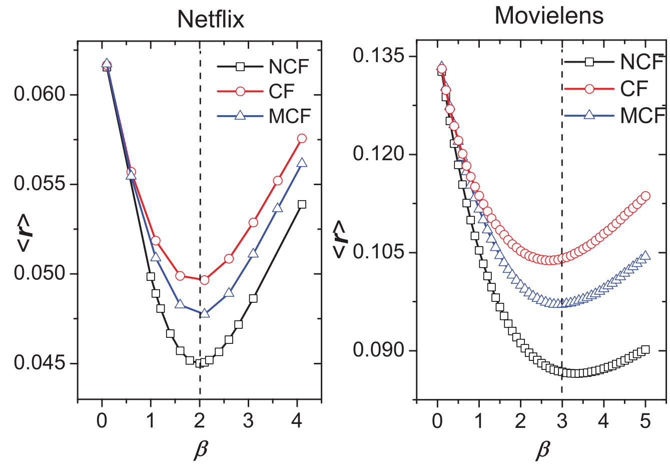

FIG. 3. (Color online) The average ranking score $\langle r \rangle$ vs β for NCF, CF, and MCF algorithms. The optimal $\beta_{\rm opt}$ of NCF for MovieLens data set, corresponding to the minimal $\langle r \rangle = 0.086$, is $\beta_{\rm opt} = 3.3$, the one for Netflix data set is $\beta_{\rm opt} = 2.0$ corresponding to $\langle r \rangle = 0.0450$. When $\beta = 1$, the algorithm degenerates to the accuracy of the CF algorithm based on the new user similarity direction. All the data points are averaged over ten independent runs with different data-set divisions.

popularity $\langle k \rangle = 237$, which is reduced by 13.8%, and the diversity $S = 0.8929$ is improved by 5.9% comparing with the ones of CF at its optimal value. When the list length $L = 10$, the popularity $\langle k \rangle$ and diversity S of NCF are reduced by 10.9% and 17.64% for the Netflix data set, from which one can find that the NCF algorithm using the new directed random walks has the capability to find the niche objects, leading to diverse recommendations.

In general, NCF outperforms CF as well as MCF in terms of the accuracy $\langle r \rangle$, diversity S, and popularity $\langle k \rangle$. However, in reality, users care only about the top part of the recommendation list. From Figs. 4(e)–4(h), one can find that, comparing with the results of CF and MCF algorithms, the precision P and recall R of NCF are also very good. When $L = 10$ with the optimal parameter corresponding to the lowest ranking score, the precision P is approximately improved 3.0% and 5.5%, and the recall R is roughly enhanced by 5.2% and 20.15% for MovieLens and Netflix data sets, respectively.

Since the similarities generated by the random-walk process from small-degree to large-degree users are larger than the ones from the opposite direction, the simulation results indicate that enhancing the small-degree users' recommendation powers increases the prediction accuracy and helps users find niche objects, leading to more diverse recommendations. Figure 5 investigates the correlation between the target user degree k_u and its neighbors' average degree $\langle k_u^n \rangle$ as well as deviation $D(k_u)$, where the target user's neighbors U_n are defined as the users who have at least one common rated object with the target user, which could be obtained from the adjacent matrix A. Denoting the user correlation matrix as $C^{\rm user}$, we have $C^{\rm user} = AA^T$. The element $C_{ij}^{\rm user}$ means the number of common rated objects between user u_i and u_j. Given a matrix $T = \{t_{ij}\} \in R^{n,n}$, with $t_{ij} = 1$ if $C_{ij}^{\rm user} > 0$, and $t_{ij} = 0$ if $C_{ij}^{\rm user} = 0$. The number of correlated neighbors k_u^c for a target user u could be given as $k_u^c = \sum_{j=1}^n t_{uj}$, then average degree $\langle k_u^n \rangle$ of correlated

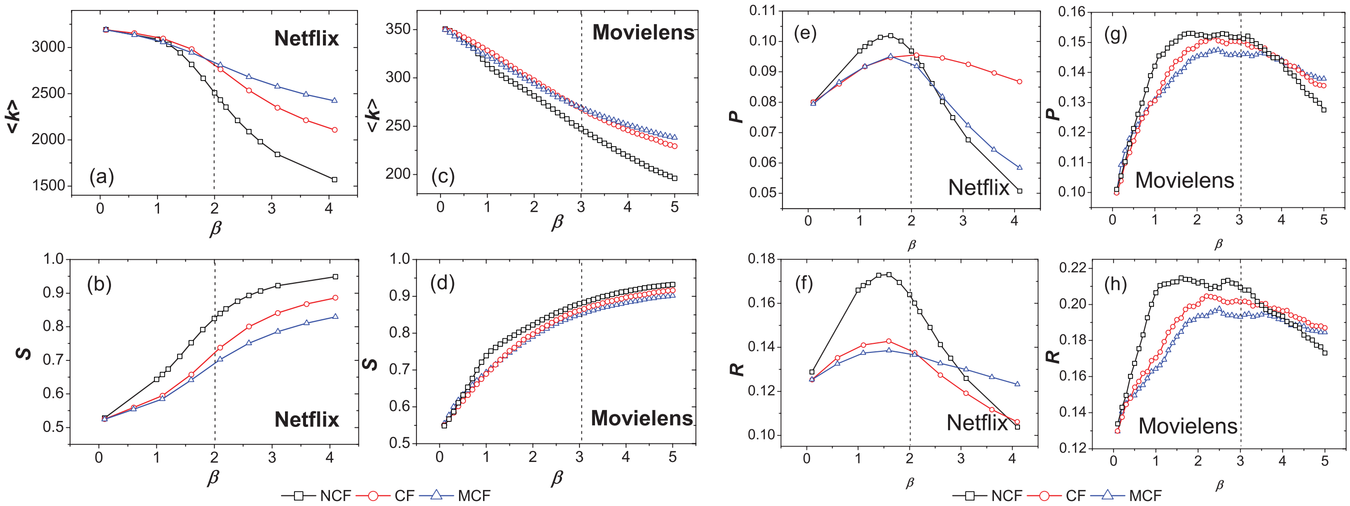

FIG. 4. (Color online) Performances of the NCF, CF, and MCF algorithms for MovieLens and Netflix data sets when recommendation lists are equal to $L = 10$. (a)–(d) Average object degrees $\langle k\rangle$ vs β and diversity S vs β. At the optimal cases, both popularity $\langle k\rangle$ and diversity S of NCF are much better than the ones of the CF and MCF algorithms. (e)–(h) Precision P and recall R vs β for Netflix and MovieLens data sets. One can find that both of P and R of NCF for MovieLens are larger than the ones of CF and MCF algorithms, while the precision P for Netflix is close to the one of the CF algorithm, and the recall R is much better than the results of the CF algorithm. All the data points are averaged over ten independent runs with different divisions of training-probe sets.

neighbors U_n is defined by

$$\langle k_u^n\rangle = \frac{1}{k_u^c}\sum_{j=1}^{n} t_{uj}k_j. \tag{14}$$

The deviation $D(k_u)$ could be given as

$$D(k_u) = \sqrt{\frac{1}{k_u^c}\sum_{j=1}^{n}\left(t_{uj}k_j - \langle k_u^n\rangle\right)^2}. \tag{15}$$

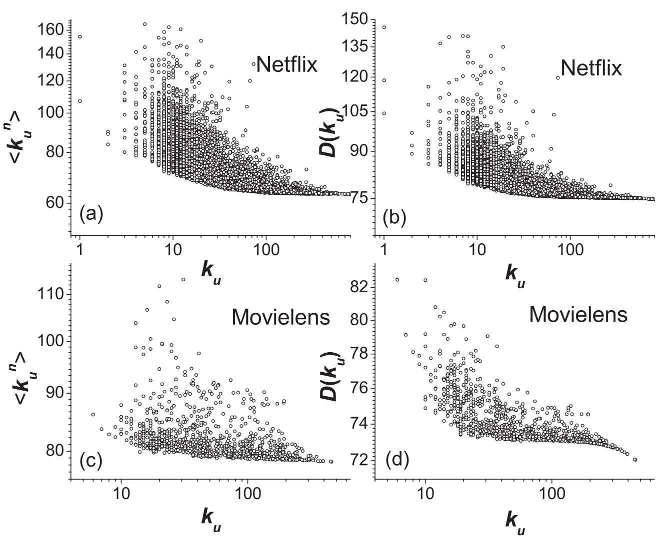

FIG. 5. (a)–(b) The average degree of the target user's neighbors $\langle k_u^n\rangle$ and the corresponding deviation $D(k_u)$ vs target user degree k_u for the Netflix data set. (c)–(d) Results for MovieLens. To small-degree users, both their neighbors' average degree $\langle k_u^n\rangle$ and the deviation D are very large. As k_u increases, $\langle k_u^n\rangle$ and $D(k_u)$ would decease correspondingly.

Figure 5 shows that when k_u is very small, both neighbors' average degree $\langle k_u^n\rangle$ and deviation $D(k_u)$ are very large, which means that for MovieLens and Netflix data sets, the small-degree users would like to commonly rate objects with small-degree users and large-degree users. As k_u increases, both $\langle k_u^n\rangle$ and $D(k_u)$ would decrease correspondingly, which means that the large-degree users only commonly rate objects with small-degree users. According to our previous analysis, if the user similarities from neighbors to the target user are enhanced, the effects of the small-degree users would be emphasized to match both large-degree and small-degree users' common and specific interests, which is the reason why directed random-walk-based user similarity is effective.

VI. EFFECTS OF DATA SPARSITY

We investigate the effects of the data sparsity on the performance. Since we focus on the similarity direction effect on the CF algorithm, we choose the classical CF algorithm for comparison. In the simulation work, we select pE edges as a training set and set the rest of the $(1-p)E$ edges as a probe set. Lower p means less information is used to generate the recommendations. The numerical results for MovieLens are shown in Fig. 6. Each point of the histogram is obtained with the optimal parameter subject to the lower ranking score. The improvement function $f(\langle r\rangle)$ of the present algorithm is defined as

$$f(\langle r\rangle) = \frac{\langle r\rangle_{\mathrm{CF}} - \langle r\rangle_{\mathrm{NCF}}}{\langle r\rangle_{\mathrm{CF}}}. \tag{16}$$

For the popularity $\langle k\rangle$ and Hamming distance S, the improvement functions are defined as

$$f(\langle k\rangle) = \frac{\langle k\rangle_{\mathrm{CF}} - \langle k\rangle_{\mathrm{NCF}}}{\langle k\rangle_{\mathrm{CF}}},\qquad f(S) = \frac{S_{\mathrm{NCF}} - S_{\mathrm{CF}}}{S_{\mathrm{CF}}}. \tag{17}$$

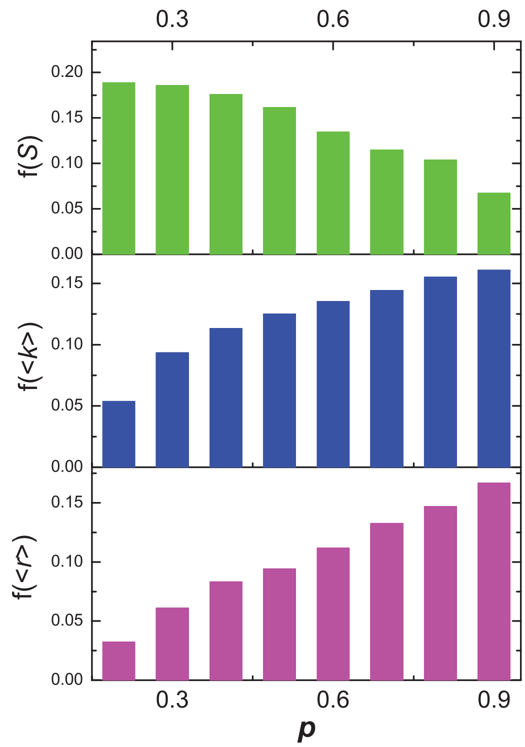

FIG. 6. (Color online) Improvements of the average ranking score $\langle r\rangle$, average objects degree $\langle k\rangle$, and diversity S to different sparsity of the training set for MovieLens data set. All the data points are averaged over ten independent runs with different data set divisions.

Figure 6 shows that the improvement of average ranking score $\langle r\rangle$ decreases as the size of the training set decreases, which may come from the fact that the number of neighbors would decrease and less information could be used to predict the target user's interests. We also found that NCF performs much better than CF for denser data sets. The improvement of diversity $f(S)$ decreases with the increasing size of the training set, and $f(\langle k\rangle)$ increases with a more denser training set, which indicates that, generally speaking, users prefer to select popular objects as they give more ratings.

VII. CONCLUSION AND DISCUSSIONS

In this paper by tuning the random-walk direction from neighbors to the target user to calculate the directed user similarity, we investigate the physics of directed random walks and their influence on the information filtering of user-object bipartite networks. Simulation results indicate that the new CF algorithm using the new directed random walks outperforms state-of-the-art CF methods in terms of the accuracy, as well as the Heter-CF algorithm in terms of diversity simultaneously. Meanwhile NCF has much better capability to present more accurate and diverse recommendations than CF algorithms whose user similarities are calculated from the target user to neighbors. The accuracies of NCF algorithms are close to the results of the hybrid algorithm [14], and the diversities are also increased dramatically.

CF algorithms are one of the most successful information-filtering algorithms and have been extensively used on many web sites, such as Netflix and Amazon. The HC model also has been successfully used for information filtering. If we suppose the user-rated specific objects are the heat source, CF algorithms are equivalent to the user-based HC model. Although random walks have been used to improve the user or object similarity measurement [10,14], the reason why directed similarity could enhance the information-filtering performance is missing. Since the idea of the CF algorithm is combing neighbors' opinions of the target user to predict his interests or habits, we always suppose that the CF algorithm would like to converge users' interests and present popular objects. But the analysis in this paper indicates that if small-degree users' recommendation powers are increased, the CF algorithm also could solve the accuracy-diversity dilemma. In most of the online social systems, the number of small-degree users is always much larger than large-degree ones. According to the random-walk-based user similarity definition, we know that similarities between small-degree users are always larger than the reversed ones. Therefore, in the CF algorithm, the opinions of the large-degree users would be recommended to most of the small-degree users, leading to lower diversity. By tuning the similarities from neighbors to the target user, we could emphasize the recommendation powers of small-degree users and enhance the accuracy and diversity simultaneously, which indicates that the similarity direction is an important factor for information filtering. Although the idea of this paper is simple, the remarkable simulation results indicate that, to generate accurate and diverse recommendations, we need only to change the direction without changing the framework of the existing CF systems.

The directed random-walk process presented in this paper indeed has been defined as a local index of similarity in link prediction [31,32], community detection [33], and so on. Meanwhile, a number of similarities, based on the global structural information, have been used for information filtering, such as the transferring similarity [12] and the PageRank index [4], communicability [34], and so on. Although the calculation of such measures is of high complexity, it is very important to the effects of directed random walks on these measures. The hybrid algorithm [14] is also a kind of item-based CF algorithm where the item similarity is measured by combining the random walk and heat conduction processes. Lü *et al.* [35] proposed an improved hybrid algorithm by embedding the preferential diffusion process into a hybrid algorithm. Qiu *et al.* [36] proposed an improved method by introducing an item-oriented function to solve the cold-start problem. In this paper we find that the direction of random walks is very important for information filtering, which may be helpful for deeply understanding the applicability of directed similarity.

ACKNOWLEDGMENTS

We thank Dr. Chi Ho Yeung for his helpful suggestions. We acknowledge GroupLens Research Group for providing us MovieLens data and Netflix Inc. for Netflix data. This work is partially supported by NSFC (10905052, 70901010, 71071098, and 71171136); J.G.L. is supported by the European Commission FP7 Future and Emerging Technologies Open Scheme Project ICTeCollective (238597), the Shanghai Leading Discipline Project (S30501), and Shanghai Rising-Star Program (11QA1404500).

[1] M. Faloutsos, P. Faloutsos, and C. Faloutsos, Comput. Commun. Rev. **29**, 251 (1999).
[2] A. Broder, R. Kumar, F. Moghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, Comput. Netw. **33**, 309 (2000).
[3] G. Adomavicius and A. Tuzhilin, IEEE Trans. Knowl. Data Eng. **17**, 734 (2005).
[4] S. Brin and L. Page, Comput. Netw. ISDN Syst. **30**, 107 (1998).
[5] J. Kleinberg, J. ACM **46**, 604 (1999).
[6] J. L. Herlocker, J. A. Konstan, K. Terveen, and J. T. Riedl, ACM Trans. Inform. Syst. **22**, 5 (2004).
[7] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, Commun. ACM **40**, 77 (1997).
[8] Z. Huang, H. Chen, and D. Zeng, ACM Trans. Inform. Syst. **22**, 116 (2004).
[9] T. Zhou, R.-Q. Su, R.-R. Liu, L.-L. Jiang, B.-H. Wang, and Y.-C. Zhang, New J. Phys. **11**, 123008 (2009).
[10] J.-G. Liu, B.-H. Wang, and Q. Guo, Int. J. Mod. Phys. C **20**, 285 (2009).
[11] J.-G. Liu, T. Zhou, H.-A. Che, B.-H. Wang, and Y.-C. Zhang, Physica A **389**, 881 (2010).
[12] D. Sun, T. Zhou, J.-G. Liu, R.-R. Liu, C.-X. Jia, and B.-H. Wang, Phys. Rev. E **80**, 017101 (2009).
[13] Y.-C. Zhang, M. Blattner, and Y.-K. Yu, Phys. Rev. Lett. **99**, 154301 (2007).
[14] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, Proc. Natl. Acad. Sci. USA **107**, 4511 (2010).
[15] J.-G. Liu, Q. Guo, and Y.-C. Zhang, Physica A **390**, 2414 (2011).
[16] X. Pan, G.-S. Deng, and J.-G. Liu, Chin. Phys. Lett. **27**, 068903 (2010).
[17] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, in *Proc. of SIGIR '99* (1999), p. 230.
[18] H. Luo, C. Niu, R. Shen, and C. Ullrich, Mach. Learn. **72**, 231 (2008).
[19] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, in *Proc. of ACM on WWW Conference* (2001), p. 285.
[20] M. Deshpande and G. Karypis, ACM Trans. Info. Syst. **22**, 143 (2004).
[21] M. Gao, Z. F. Wu, and F. Jiang, Info. Proc. Lett. **111**, 440 (2011).
[22] E. A. Leicht, P. Holme, and M. E. J. Newman, Phys. Rev. E **73**, 026120 (2006).
[23] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, Phys. Rev. E **76**, 046115 (2007).
[24] B. Travencolo and L. da F. Costa, Phys. Lett. A **373**, 89 (2008).
[25] B. A. N. Travenolo, M. P. Viana, and L. da F. Costa, New J. Phys. **11**, 063019 (2009).
[26] M. P. Viana, J. B. Batista, and L. da F. Costa, e-print arXiv:1101.5379.
[27] T. Zhou, L. L. Jiang, R. Q. Su, and Y.-C. Zhang, Europhys. Lett. **81**, 58004 (2008).
[28] M. B. Díaz, M. A. Porter, and J.-P. Onnela, Chaos **20**, 043101 (2010).
[29] J. P. Onnela and F. Reed-Tsochas, Proc. Natl. Acad. Sci. USA **107**, 18375 (2010).
[30] J.-G. Liu, T. Zhou, and Q. Guo, Phys. Rev. E **84**, 037101 (2011).
[31] L. Lü and T. Zhou, Europhys. Lett. **89**, 18001 (2010).
[32] T. Zhou, L. Lü, and Y.-C. Zhang, Eur. Phys. J. B **71**, 623 (2009).
[33] Y. Pan, D.-H. Li, J.-G. Liu, and J.-Z. Liang, Physica A **389**, 2849 (2010).
[34] E. Estrada and N. Hatano, Phys. Rev. E **77**, 036111 (2008).
[35] L. Lü and W. Liu, Phys. Rev. E **83**, 066119 (2011).
[36] T. Qiu, G. Chen, Z.-K. Zhang, and T. Zhou, Europhys. Lett. **95**, 58003 (2011).