Information filtering in evolving online networks

Bo-Lun Chen[a,b], Fen-Fen Li[a,*], Yong-Jun Zhang[a], Jia-Lin Ma[a]

[a] *College of Computer Engineering, Huaiyin Institute of Technology, Huaian 223300, China*
[b] *Department of Physics, University of Fribourg, Chemin du Musée 3, CH-1700 Fribourg, Switzerland*

Recommender systems use the records of users' activities and profiles of both users and products to predict users' preferences in the future. Considerable works towards recommendation algorithms have been published to solve the problems such as accuracy, diversity, congestion, cold-start, novelty, coverage and so on. However, most of these research did not consider the temporal effects of the information included in the users' historical data. For example, the segmentation of the training set and test set was completely random, which was entirely different from the real scenario in recommender systems. More seriously, all the objects are treated as the same, regardless of the new, the popular or obsoleted products, so do the users. These data processing methods always lose useful information and mislead the understanding of the system's state. In this paper, we detailed analyzed the difference of the network structure between the traditional random division method and the temporal division method on two benchmark data sets, Netflix and MovieLens. Then three classical recommendation algorithms, Global Ranking method, Collaborative Filtering and Mass Diffusion method, were employed. The results show that all these algorithms became worse in all four key indicators, ranking score, precision, popularity and diversity, in the temporal scenario. Finally, we design a new recommendation algorithm based on both users' and objects' first appearance time in the system. Experimental results showed that the new algorithm can greatly improve the accuracy and other metrics.

1. Introduction

Many natural and social systems can be described by networks and graphs [1,2]. Among these networks, many of them emerge a natural bipartite structure, i.e. a network consisting of two separated node sets and links between the two node sets. Examples of this kind of networks are scientific collaboration network [3,4], P2P Internet formed by computer terminals and data [5]; cooperative network formed by actors and their films [6,7]; Shares Network formed by the investors and companies [8,9]; the activity network formed by the members of the club and their activities [10]; the audience with the songs network [11]; disease-gene networks [12], and so on.

In e-commercial systems, the bipartite network is an important tool to characterize the relations between users and objects. Based on the user-object bipartite network, many recommendation algorithms are developed. The existing recommendation algorithms can be divided into four main categories: collaborative filtering algorithm [13,14], association rule-based algorithm [15], content-based algorithm [16,17] and resource allocation algorithm [18–20].

The main idea of the collaborative filtering is to do a recommendation based on the similarity of nodes, which can be further classified as user-based similarity or object-based similarity. They can help customers quickly to find what they want. Lu et al. [21] carried out a research going through the latest application of the recommender system by classifying the applications into eight types, including e-group activities, e-resource services, e-tourism, e-learning, e-library, e-commerce/e-shopping, e-business and e-government. Besides, a summary is also made about relevant recommendation techniques of each type. This survey can greatly help the researchers to promote their understandings about the application development of recommender system. In addition, Yera et al. [22] presented a review about the application of fuzzy tools in this research. According to his research, application of fuzzy tools is mainly used to detect more exclusively studied topic and also research gaps and hence to make proper suggestions for the subsequent researches to promote the further development of the fuzzy-based recommender system. Particularly, an analysis is made in his research about how the application areas, employed dataset, evaluation strategies and key features of the Thomson Reuters Web of Science have been applied to realize such an aim.

[*] Corresponding author.
E-mail addresses: chenbolun@hyit.edu.cn (B.-L. Chen), lifenfen@hyit.edu.cn (F.-F. Li).

Despite its success, collaborative filtering suffers from data sparsity problem [23,24]. The association rule-based recommendation algorithm can find the correlations of different products in the sales process and has been successfully applied in many web sites. This method does not require domain knowledge to discover new points of interest. However, the critical bottleneck of the algorithm is that it's difficult to extract rules effectively because the time complexity of the association rule-based algorithm is very high, whereas the diversity index of the recommendation is low. Content-based recommendation uses the historical information (such as rating scores, sharing activities, and documents collecting records) to construct user preferences file, then calculates the similarity between recommended objects and user preferences, and finally recommends the most similar objects to the target user. The shortcoming of this method is that it is difficult to acquire useful information from non-textual data format, such as images, music, videos etc. Therefore, the content-based method may fail to find the interest of the users without deviation. Resource allocation methods are designed by physicists. These methods compute the users' or/and objects' similarity through diffusion processes on user-object bipartite network and recommend objects to the target user based on the final resource the objects received. These physical based algorithms are highly accurate and efficient and could be used to resolve the apparent accuracy–diversity dilemma when combined in an elegant hybrid with an mass (resource) diffusion [18] and heat conduction algorithm [20], to cope with the cold start problems in recommendation [25], to avoid congestion in the stocks is limited [26], etc.

2. Related work

In recent years, some researchers also tried to integrate the temporal information into the recommendation algorithms [27,28]. These temporal effects include engagement of new users, seasonal effects, user preference drifts and shifts and other phenomena alike. These phenomena can have a continuous impact over the underlying relations between the items and users, which is also what recommendation algorithms attempts to capture.

Revamping two dominating collaborative filtering algorithms, Koren et al. [29] presented a model tracking the time changing behaviors in order to distill the longer-term trends from the noisy patterns. Xiong et al. [30] proposed a Bayesian probabilistic tensor factorization (TF) model based on the continuous time. Liu et al. [31] studied the effect of temporal information on the heat-conduction algorithm by gradually expanding time window. Liu et al. [32] found that both temporal attenuation and diversion delay play key roles in recommender system when link weight is taken into consideration. Then they combined these two temporal factors with users' life spans to construct a time-weighted network (TWN) model and resource allocation process was applied.

There are several other examples of algorithms that use time as context available in the literature on context-aware recommendation. For instance, Maioetal et al. [33] promoted that the time-aware collaborative filtering can be adopted to evaluate what the users might hold interest on when they browse through Twitter. Text analysis service is adopted in this approach to annotate the content of Tweets semantically to track the notions based on the frequencies of being posted and forwarded along the time. Sara et al. [34] made a study to get the contextual recommendation customized through a time-awareness system.

Even though many recommendation algorithms had been designed and in some of them the temporal information were applied to improve the recommendation accuracy, one important issue is still being serious overlooked: some of temporal methods are done based on randomly divided training set and probe set [18,19]. The main task of recommender system is to predict future links of each target user based on historical rating records [31,32, 35–37], in which the whole data set is divided into a training set and a test set and only the links in training set is known before recommendation. However, in most of the previous studies, as far as we know, the test data set is sampled randomly without consider the temporal order of the links, leading to a logical disorder in the recommendation process, i.e., predicting the past links in test set on the basis of the future links in training set. Therefore, the real performance of the existing recommendation algorithms, such as the famous Global Ranking Method (GRM) [18], User-based Collaborative Filtering (UCF), and Mass Diffusion method (MD), is actually not yet fully understood. In this paper, we reexamine some representative recommendation algorithms with the temporal division method, i.e. the observed links and unknown future links are divided strictly based on their temporal order. We find that most recommendation algorithms have much lower accuracy when they are applied to temporal division training set and test set, than random division method. After careful analysis, we find this is because there are more novel objects, with a little historical links in training set, need to be recommended in temporal division set, comparing with the random division set.

Additionally, to solve this problem, we present a new recommendation algorithm to improve the recommendation accuracy by considering the first appearance time of the objects and users in system.

3. Problem formulation and evaluation methods

A bipartite network model gives us a clear vision when we deal with the problem involving two different types of nodes. In network based recommendation algorithm, data sets usually represented by an undirected bipartite network $G(U, O, E)$, where U is the set of users, O is the set of objects and E is the set of links between them. The links between uses and objects can also be represented by an adjacent matrix $A_{|U|\times|O|} = \{a_{i\alpha}\}$, where $a_{i\alpha} = 1$ if user u_i selected object o_α and $a_{i\alpha} = 0$ otherwise.

The purpose of a recommendation algorithm is to assign a score, $Score(i, \alpha)$, to each unconnected pair of user and object (i, α). This score value may be understood as a kind of proximity, which associated with their connection probability. That is to say, for a pair of unconnected nodes (i, α), the larger the $Score(i, \alpha)$ is, the higher the probability there will exist a link between user i and object α. In most of the previous studies, the links in E are randomly divided into two subsets: training set E^T, which is treated as known information, and probe set E^P, which is unknown before recommendation, to test the recommendation performance of an algorithm, e.g., accuracy and diversity. It's clear that $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$.

In principle, a recommendation algorithm will provides each target user with an descending ordered list of objects that the target user did not chosen before. For each recommended objects in the list, if there exist a link between the object and target user in probe set, we call it a hitting. Note that, only the user who have at least one selected objects in training set and at least one connected object in probe set will be considered when computing algorithms' performance. In this work, four popular used metrics are employed to test the effectiveness of the algorithms, *Ranking Score*, *Precision*, *Popularity* and *Diversity*.

Ranking Score (RS) is one of the most important accuracy metrics which measures the ability of recommendation algorithm to assign more score to users' preferable objects than irrelevant ones. Considering a target user i and its recommendation list, the ranking score of each selected object α in probe set is $RS_i\alpha = rank_\alpha/(|O| - k_i^T)$ where $rank_\alpha$ is the rank of the object α in the descending ordered recommendation list of the target user i and

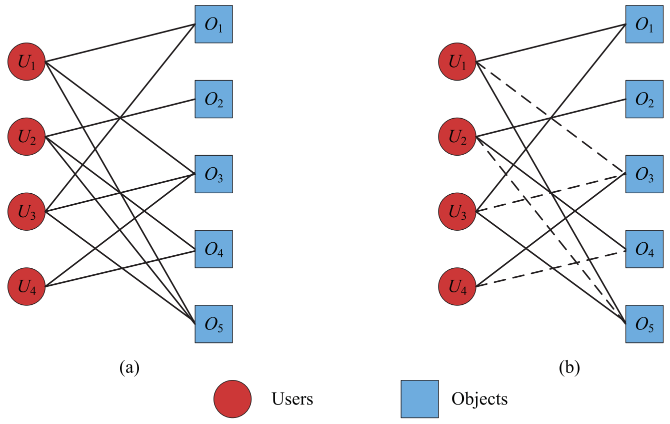

Fig. 1. An illustration about the calculation of the four metrics.

k_i^T is the degree of target user i in training set. Thus, the integrated ranking score of the whole system is the average ranking score over all users and objects pairs in probe set, reads

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

Note that, the ranks of the objects with same scores are the same and equal to their median. It can clear that prediction result with higher accuracy will get higher ranking score.

Given the ranking of the non-observed links, the *Precision* (P) is defined as the ratio of relevant objects selected to the total number of objects selected. That is to say, if we take the $Top - L$ links as the predicted ones, among which hi links are right, then the *Precision* can be expressed as:

$$P = \frac{1}{n} \sum_{i=1}^{n} \frac{h_i}{L} \tag{2}$$

Clearly, higher precision means higher prediction accuracy.

The metric *Popularity* (I) measures the average degree of the objects in the recommendation list. It is hard for the users to find the relevant but unpopular objects. Therefore, a good recommender system should prefer to recommend small degree objects. The metric *Popularity* (I) can be expressed as:

$$I_i(L) = \frac{1}{L} \sum_{\alpha\in O^i} k_\alpha \tag{3}$$

where O^i represents the recommendation list for user i, k_α represents the degree of the object α. A low mean popularity $I(L)$ for the whole system indicates a high novel and unexpected recommendation of objects.

The *Diversity* (D) mainly considers how users' recommendation lists are different from each other. Here, we measure it by the Hamming distance. We denote $C_{ij}(L)$ as the number of common objects in the top-L place of the recommendation list of user i and j, their hamming distance can be calculated as:

$$D_{ij}(L) = 1 - \frac{C_{ij}(L)}{L}. \tag{4}$$

$D_{ij}(L)$ is between 0 and 1, which are respectively corresponding to the cases where user i and j have the same or an entirely different recommendation list. By averaging $D_{ij}(L)$ over all pairs of users, we obtain the mean hamming distance $D(L)$. The more the recommendation list differs from each other, the higher the $D(L)$ is.

Fig. 1 gives an example of how to calculate the four metrics. Fig. 1(a) shows a complete network structure which includes four users and five objects. In the whole graph, we can see eleven existent links and nine nonexistent links. To test the algorithm's accuracy, we need to choose some existent links as probe set.

For instance, we select (U_1, O_3), (U_2, O_5), (U_3, O_3) and (U_4, O_4) as probe links, which are presented by dotted lines in Fig. 1(b). Then, the remaining seven links constitute the training set. An algorithm can only make use of the information contained in the training set (presented by solid lines in Fig. 1(b)). If an algorithm assigns scores of all non-observed links as $Score(U_1, O_2) = 0.4$, $Score(U_1, O_3) = 0.8$, $Score(U_1, O_4) = 0.6$, $Score(U_2, O_1) = 0.4$, $Score(U_2, O_3) = 0.8$, $Score(U_2, O_5) = 0.7$, $Score(U_3, O_2) = 0.4$, $Score(U_3, O_3) = 0.8$, $Score(U_3, O_4) = 0.6$, $Score(U_4, O_1) = 0.4$, $Score(U_4, O_2) = 0.5$, $Score(U_4, O_4) = 0.6$, $Score(U_4, O_5) = 0.7$. Then, we should sort the scores in descending order for each target user.

To calculate RS, the RS value of U_1 equals $1/(5-2) = 1/3$, the RS value of U_2 equals $2/(5-2) = 2/3$, the RS value of U_3 equals $1/(5-2) = 1/3$, the RS value of U_4 equals $2/(5-1) = 1/2$. Hence, the RS value of this algorithm equals $(1/3 + 2/3 + 1/3 + 1/2)/4 = 11/24$.

To calculate P, here, we set L as 2. Then, the P value of U_1 equals 1/2, the P value of U_2 equals 1/2, the P value of U_3 equals 1/2, the P value of U_4 equals 0. Hence, the P value of this algorithm equals $(1/2 + 1/2 + 1/2 + 1/2)/4 = 1/2$.

To calculate D, here, we set L as 3. Then, the D value of (U_1, U_2) equals $1 - 1/3 = 2/3$, the D value of (U_1, U_3) equals $1 - 3/3 = 0$, the D value of (U_1, U_4) equals $1 - 2/3 = 1/3$, the D value of (U_2, U_3) equals $1 - 1/3 = 2/3$, the D value of (U_2, U_4) equals $1 - 1/3 = 2/3$, the D value of (U_3, U_4) equals $1 - 2/3 = 1/3$. Hence, the D value of this algorithm equals $(2/3 + 0 + 1/3 + 2/3 + 1 + 1/3)/6 = 1/2$.

4. Data and methods

4.1. Data and the correlation of objects' degree

In this paper, we use two standard data sets which have been widely used to examine the performance of recommendation algorithms [27,28]. The first one is the Netflix with 1891 movies (objects) and 2294 users (http://www.Netflixprize.com/). Users rate movies from 1 (worst) to 5 (best). Consistent with the literature, we consider the ratings higher than 2 as a link. Finally, 59464 links remain in the network. The second one is the Movielens data which is a random sample of the whole records of users ratings during the seven-month period from 19 September 1997 to 22 April 1998 in Movielens.com (http://www.grouplens.org/). It consists of 943 users, 1682 movies, and 100000 links. Like Netflix, Movielens is also based on a 5-star rating systems. With the same rating filtering process as Netflix, we obtain 82520 links in Movielens data.

Because the links of the objects may both appear in the training set and in the test set, so we use the number of the objects links occurred in the training set as the abscissa and the number of the objects links occurred in the test set as the ordinate. We set the ratio of training set is 90% and draw the correlation of the objects' degree in the data set both divided by random and by time. The result of Netflix is shown in Fig. 1, the result of Movielens is shown in SI.

Form Fig. 2, we can see that NP and NT almost show a linear relationship. It is because that the links are selected as training set and probe set by equal probability. But in the condition of division data sets by temporal information, the distribution of points are more discrete. There exist that some objects' degrees are smaller in the training set but bigger in the probe set, and some objects'

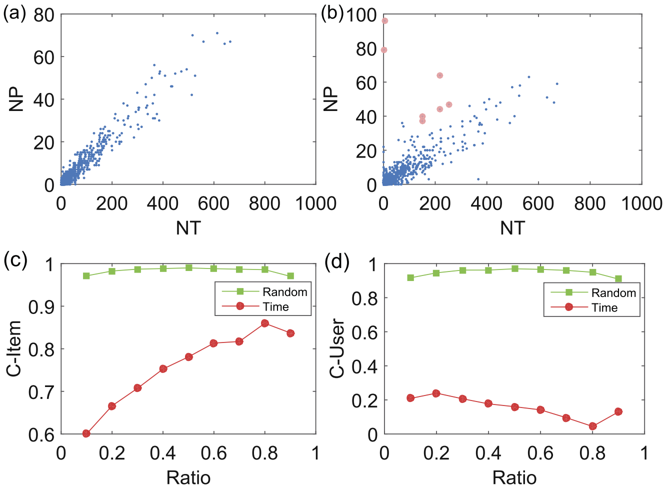

Fig. 2. The analysis of the objects' and users' degree in training set and probe set, the Netflix case. The scatter plots in panels (a) and (b) are the objects's degree in training set versus probe set when the data set is divided randomly and according temporal information, respectively. The ratio of training set is 0.9. The highlight points in (b) are the cold-start objects and obsolete objects, which are indispensable in real recommender system. Panel (c) is the correlation of objects' degree in training set and probe set versus the changes of the ratio of training set. The green line and red line correspond to the sets that is divided randomly and divided according the temporal sequence, respectively. Panel (d) is the same as (c), but for the correlation of users' degree. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

degrees are bigger in the training set but smaller in the probe set. Even more, some objects' degrees are zero in the training set, but non-zero in the probe set that is the cold start problem. Those objects are highlighted in the Fig. 2. We also can see that $C - Item$ and $C - User$ almost near 1 in the data sets divided by random as the changing of the training set's ratio.

4.2. The age of the object

We further move to study the attributive character of the object and give the following defines firstly.

Define 1. The age of the object (T_{AO}): we supposed that the object receives the first link is the birthday, we set it T_{BO}. We set the ending time of the dataset is T_E, so the age of the object is $T_{AO} = T_E - T_{BO}$.

Define 2. The age of the object(T_{AU}): we supposed that the user receives the first link is the birthday, we set it T_{BU}. We set the ending time of the dataset is T_E, so the age of the user is $T_{AU} = T_E - T_{BU}$.

In the Netflix, we set the age interval of objects $\Delta T = 100$ and in this way the whole objects could be divided into 22 groups. Analogously, we assume that age interval ΔT=100 and in this way the whole objects could be divided into 22 groups in the Movielens. We count the number of objects in every age's range both in the whole data sets but in the probe sets. In the probe sets, we considered the random division of data sets and data sets divided by the temporal information in both cases. The results of Netflix and Movielens are shown in Fig. 2 and SI, respectively.

Form Fig. 3, we can see that in the Netflix, the trend of the number in the probe sets divided by random is similar to the number in the whole data sets and the number of young objects' age based on division data sets by temporal information are more than the number of young objects' age based on division data sets by random. In contrast, the number of old objects' age based on division data sets by temporal information are smaller than the number of old objects' age based on division data sets by random. It indicates that we can get different probe sets according to different division ways. The result of Movielens is shown in SI.

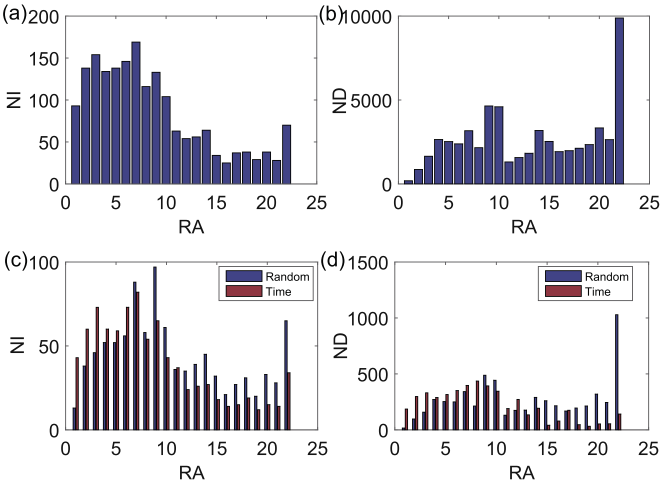

Fig. 3. The evolving of objects' quantity and the average degree with the increasing of objects' age in the Netflix data set. Panels (a) and (b) are the objects' the quantity and the average degree evolving, respectively, in the whole data set of Netflix. Panels (c) and (d) are the evolving analysis in the probe set of Netflix. The same analysis of Movielens data set is shown in SI.

4.3. The analysis of data sets divided by random and time

The main task of recommender system is to predict a future link of a user based on a historical rating link, but we used to divide the data set into two parts by random that caused the links occurred at the latest time as the training data, and we use the links occurred at the latest time to predict the links occurred at the past time. In order to reveal the effect of users' online behavior patterns on recommender system, we divided the data set by time. The records that occurred at the latest time are set testing data, and others as training data. The ratio of training set increases from 10% to 90%. We use three different recommendation methods such as user based collaborative filtering(UCF), global rank method(GRM) and mass diffusion algorithm(MD) on the different sizes of training data set divided by random and temporal information. The four materials values on Netflix are shown in Fig. 3. The result of four materials values on Movielens are shown in SI.

From Fig. 4, we can see that different recommendation methods present the same trend with the increase of the *Ratio* whether division data set by random or by temporal information. The *RankingScore* of all methods are getting worse when the data set divided by temporal information, it indicates that the number of young links is more than that of old links, and those young links are not correct recommendation. Similarly, the *Precision* of all methods are worse, because some new links are used to predict the old links in the condition of division data set by temporal information. Finally, we average every material value in different ratio of training data sets, the result is shown in Table 1.

From Table 1, we can see that the all metrics almost become worse both in Netflix and Movielens except diversity when the data set divided by temporal information. Specifically, compared to the random data division case, the ranking score RS is increased by 74.09%, 50.70%, 49.27% in Netflix and 35.43%, 30.67%, 31.98% in Movielens according to three different methods. The precision P, popularity I and hamming distance D of different methods are also shown in Table 1. The results of precision P in Table 1 confirm our finding that the temporal data division could result in a lower recommendation accuracy. Compared with the precision in the random data division case, P in the temporal data division is decreased by 27.10%, 35.82%, 45.25% in Netflix and 32.65%, 38.64%, 46.76% in Movielens. These results indicate that temporal data division mechanism can indeed reduce the general recommendation accuracy.

Table 1
The average value of all metrics in different ratio of training data set.

		GRM		UCF		MD	
		Random	*Time*	*Random*	*Time*	*Random*	*Time*
Netflix	RS	**0.108**	0.186	**0.121**	0.183	**0.116**	0.173
	P	**0.165**	0.107	**0.170**	0.118	**0.175**	0.119
	I	**290.0**	290.9	**250.7**	271.0	**242.6**	265.2
	D	0.203	**0.355**	**0.590**	0.542	**0.665**	0.616
Movielens	RS	**0.158**	0.201	**0.146**	0.198	**0.135**	0.196
	P	**0.282**	0.190	**0.340**	0.209	**0.372**	0.198
	I	**204.2**	205.7	**187.3**	189.4	**180.1**	183.1
	D	0.300	**0.565**	0.625	**0.691**	0.7154	**0.739**

Here we set the recommendation list $L = 10$.

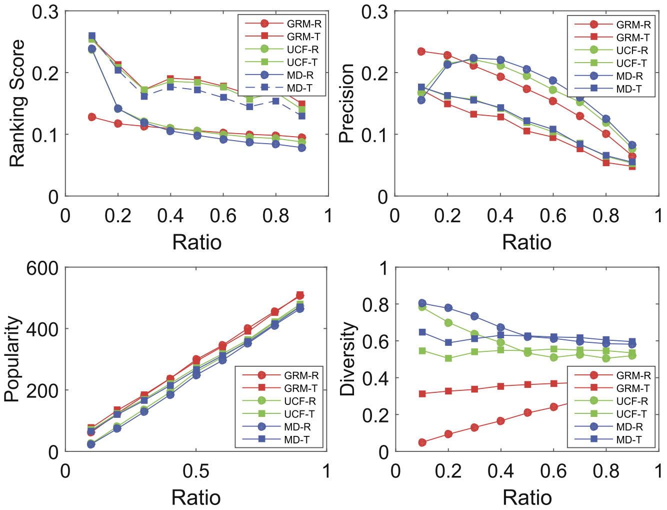

Fig. 4. The performance of three classic recommendation algorithms, global ranking method (GRM), user-based collaborative filtering (UCF) and mass diffusion method (MD), both on the data set divided randomly and by time sequence. Four key criterions of recommender systems, ranking score, precision, popularity and diversity, are employed in this figure. The horizontal axis is the ratio of training set, and the length of the recommendation list is $L = 10$.

Through the series of analysis, we can see that recommendation algorithm can achieve better recommendation results in the case of division data sets by random, but when we consider the temporal information, the performance of the algorithm has dropped, so we need to present a new algorithm that can achieve better results when the data sets are divided by temporal information.

4.4. Algorithm of AD model

The basic idea of our algorithm is: firstly, we set unique θ to every target user according to their ages. The older users in real systems show a small θ while the younger users are with higher θ. Second, we rank the object according to their ages and degrees. We put the objects which age is smaller and degree is bigger on the forefront of the queue $L1$. In the next, we build another queue $L2$ by mass diffusion algorithm. So, θ is the parameter to adjust the two lists. Actually, the younger users don't have much experience in exploring new objects, they are more likely to conservatively choose objects in the group which they are already familiar with. On the contrary, the older user in real systems inclines to search and try unpopular objects. So, if the target user is older, the weight of $L1$ is more than $L2$, it means that we prefer to recommend to their younger and unpopular objects.

The framework of our AD algorithm for recommender system is as follows.

Algorithm AD_RS (AD for recommender system)
Input: A: The adjacency matrix of network;
Output: *Score*: The score matrix;

Begin

Step 1: We set unique θ to every target user according to formula (5):

$$\theta = \frac{\max(UserAge) - TragetUserAge}{\max(UserAge) - \min(UserAge)} \tag{5}$$

Step 2: We set $AgeObject_i$ is the age of the O_i and $DegreeObject_i$ is the degree of O_i. We calculate the value of every object by formula (6) and sort their value from small to large. We set the list as $L1$. Next, we use the ranking replace to the recommended value in the $L1$.

$$value = \frac{AgeObject_i}{DegreeObject_i} \tag{6}$$

Step 3: We get a recommended list by Mass Diffusion algorithm. We set the list as $L2$ and sort their value from large to small. Next, we use the ranking replace to the recommendation value in the $L2$.

Step 4: We generate a new recommended list to target user by formula (7) and use the *similarity* as the score matrix.

$$similarity = (1 - \theta) * \log_2 L1 + \theta * \log_2 L2 \tag{7}$$

Step 5: We sort the similarity from small to large, S is the recommended list for all user.

End

5. Results

In this section, we empirically demonstrate the effectiveness of AD algorithm on two real world networks. We also compare its performance against the traditional recommendation algorithms such as the global rank method (GRM), user based collaborative filtering (UCF), mass diffusion(MD), tensor factorization model (TF), time-weighted network model (TWN). We focus on the ranking score, precision, popularity and diversity of the algorithm. In our experiments, we set the ratio of training set increases from 10% to 90%. The result of different recommendation methods on different sizes of the training data set are shown in Fig. 5. We calculate the average value of all metrics, the result is shown in Table 2.

Accuracy is always the first consideration in evaluating a recommendation algorithm's performance. Comparing the result from the six methods, we can see that the AD outperforms the other five algorithms on ranking score and the average value of precision of the AD outperforms other three on the datasets Netflix and Movielens. For instance, compared with that of TF, the average ranking score RS can be reduced by 12.12% for Netflix, 2.11% for Movielens. When this comparison is against GRM, the reductions are further enlarged, to 22.61% for the Netflix, 7.70% for the Movielens. Besides, compared with that of TF, the average value of precision P of the AD can be increased by 4.20% for Netflix, 3.61% for Movielens. When it compared with that of GRM, the average

Table 2
The average value of all metrics in different ratio of training data set.

		GRM	UCF	MD	TF	TWN	AD
Netflix	RS	0.188	0.183	0.174	0.165	0.156	**0.145**
	P	0.107	0.118	0.118	0.119	0.122	**0.124**
	I	290.9	271.0	265.2	262.5	259.6	**256.2**
	D	0.355	0.542	0.616	0.606	0.618	**0.626**
Movielens	RS	0.201	0.198	0.196	0.190	0.188	**0.186**
	P	0.190	0.201	0.198	0.194	0.196	**0.201**
	I	201.7	189.4	183.1	182.7	179.9	**176.1**
	D	0.565	0.691	**0.756**	0.638	0.649	0.667

Here we set the recommendation list $L = 10$.

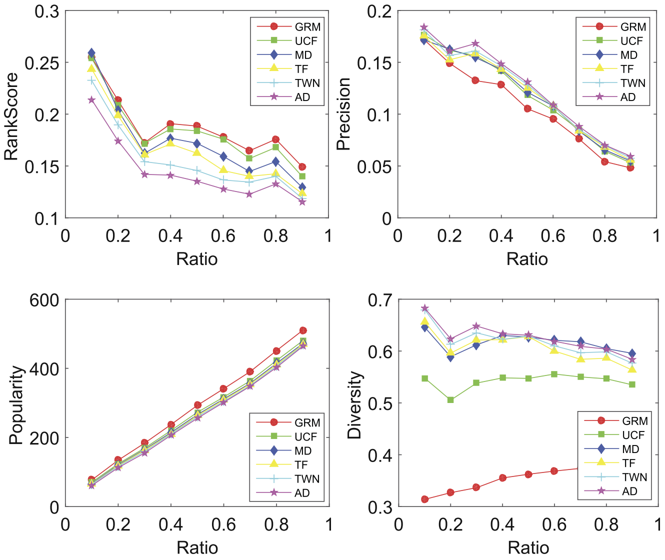

Fig. 5. Comparisons of average Ranking Score, Precision, Popularity and Diversity between the different algorithms in different sizes of training set on the Netflix. Here we set the recommendation list $L = 10$.

value of precision of the AD can be increased by 16.39% for Netflix, 5.78% for Movielens. The same effect is compared with the rest of the two algorithms. Generally, the AD can provide the best or very nearly the best accuracy, as measured by the ranking score and precision.

Besides accuracy, popularity I and diversity D are two other important metrics. From the result, we can see that the average value of popularity of the AD outperforms other five on different data sets, specifically, compared with that of TWN, the average popularity I can be reduced by 1.31% for Netflix, 2.11% for Movielens. When it compared with that of GRM, the average popularity I can be reduced by 11.91% for Netflix, 12.71% for Movielens. Besides, the average value of diversity of the AD outperforms other five on the Netflix.

6. Conclusion

With the large amount of network data become readily available in electric form today, recommender system has become a popular subarea in data mining. The aim of the traditional recommendation methods is to improve the accuracy of the recommendation, they all ignore the division of the data sets which is the basic problem of the recommendation. In this paper, we analyzed the distinctiveness between the random data division and temporal data division. We find that the distribution of the objects' degree correlation is non-linear, that means that there exist that some objects' degree are smaller in the training set but bigger in the probe set, and some are bigger in the training set but smaller in the probe set. Even more, some objects' degree are zero in the training set, but non-zero in the probe set. Then we find the effect of many methods getting worse when the data sets are divided by temporal information. So we present a new recommendation model in which we set weight to each user based on the temporal information and considered the degree and temporal information of the objects. Our experimental results show that it can obtain higher quality results on the bipartite networks.

Acknowledgements

This research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61602202, 61379066, Natural Science Foundation of Jiangsu Province under contracts BK20160428, BK20161302, 533 talent project in Huaian. Xiao-Long Ren, Matus Medo, An Zeng also supported this work.

Supplementary material

Supplementary content related to this article has been published online at https://doi.org/10.1016/j.physleta.2017.11.027.

References

[1] R.N. Lichtenwalter, J.T. Lussier, N.V. Chawla, New perspectives and methods in link prediction, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 243–252.
[2] L. Lü, T. Zhou, Link prediction in complex networks: a survey, Phys. A, Stat. Mech. Appl. 390 (6) (2011) 1150–1170.
[3] J. Liu, Y. Dang, Z. Wang, et al., Relationship between the in-degree and out-degree of WWW, Phys. A, Stat. Mech. Appl. 371 (2) (2006) 861–869.
[4] R. Guimera, M. Sales-Pardo, Missing and spurious interactions and the reconstruction of complex networks, Proc. Natl. Acad. Sci. 106 (52) (2009) 22073–22078.
[5] A. Papadimitriou, P. Symeonidis, Y. Manolopoulos, Fast and accurate link prediction in social networking systems, J. Syst. Softw. 85 (9) (2012) 2119–2132.
[6] T. Hossmann, G. Nomikos, T. Spyropoulos, et al., Collection and analysis of multi-dimensional network data for opportunistic networking research, Comput. Commun. 35 (13) (2012) 1613–1625.
[7] K. Jahanbakhsh, V. King, G.C. Shoja, Predicting missing contacts in mobile social networks, Pervasive Mob. Comput. 8 (5) (2012) 698–716.
[8] Y. Sun, R. Barber, M. Gupta, et al., Co-author relationship prediction in heterogeneous bibliographic networks, in: 2011 International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2011, pp. 121–128.
[9] X. Li, H. Chen, Recommendation as link prediction in bipartite graphs: a graph kernel-based machine learning approach, Decis. Support Syst. 54 (2) (2013) 880–890.
[10] Z. Huang, D.K.J. Lin, The time-series link prediction problem with applications in communication surveillance, INFORMS J. Comput. 21 (2) (2009) 286–303.
[11] H. Liu, Uncovering the network evolution mechanism by link prediction, Sci. Sin. Phys. Mech. Astron. 41 (2011) 816.
[12] D. Lin, An information-theoretic definition of similarity, ICML 98 (1998) 296–304.
[13] D. Goldberg, D. Nichols, B.M. Oki, et al., Using collaborative filtering to weave an information tapestry, Commun. ACM 35 (12) (1992) 61–70.
[14] J.B. Schafer, D. Frankowski, J. Herlocker, et al., Collaborative Filtering Recommender Systems. The Adaptive Web, Springer, Berlin–Heidelberg, 2007, pp. 291–324.
[15] J. Li, Y. Xu, Y. Wang, et al., Strongest association rules mining for efficient applications, in: 2007 International Conference on Service Systems and Service Management, IEEE, 2007, pp. 1–6.

[16] F.H. Wang, S.Y. Jian, An effective content-based recommendation method for Web browsing based on keyword context matching, J. Inform. Electron. 1 (2) (2006) 49–59.

[17] C. Wartena, W. Slakhorst, M. Wibbels, Selecting keywords for content based recommendation, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 1533–1536.

[18] T. Zhou, J. Ren, M. Medo, et al., Bipartite network projection and personal recommendation, Phys. Rev. E 76 (4) (2007) 046115.

[19] Y.C. Zhang, M. Blattner, Y.K. Yu, Heat conduction process on community networks as a recommendation model, Phys. Rev. Lett. 99 (15) (2007) 154301.

[20] T. Zhou, Z. Kuscsik, J.G. Liu, et al., Solving the apparent diversity–accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. 107 (10) (2010) 4511–4515.

[21] J. Lu, D. Wu, M. Mao, et al., Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.

[22] R. Yera, Martinez L. Fuzzy, Tools in recommender systems: a survey, Int. J. Comput. Intell. Syst. 10 (1) (2017) 776–803.

[23] J. Bobadilla, F. Ortega, A. Hernando, et al., Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[24] H. Ma, T.C. Zhou, M.R. Lyu, et al., Improving recommender systems by incorporating social contextual information, ACM Trans. Inf. Syst. 29 (2) (2011) 9.

[25] Z.K. Zhang, C. Liu, Y.C. Zhang, et al., Solving the cold-start problem in recommender systems with social tags, Europhys. Lett. 92 (2) (2010) 28002.

[26] X. Ren, L. Lü, R-R. Liu, et al., Avoiding congestion in recommender systems, New J. Phys. 16 (6) (2014) 63057.

[27] P. Campos, G. Diezf, I. Cantador, Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols, User Model. User-Adapt. Interact. 24 (1–2) (2014) 67–119.

[28] J. Vinagre, A.M. Jorge, J. Gama, An overview on the exploitation of time in collaborative filtering, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 5 (5) (2015) 195–215.

[29] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (2010) 89–97.

[30] L. Xiong, X. Chen, T.K. Huang, et al., Temporal collaborative filtering with bayesian probabilistic tensor factorization, SDM 10 (2010) 211–222.

[31] Q. Guo, W.J. Song, L. Hou, et al., Effect of the time window on the heat-conduction information filtering model, Phys. A, Stat. Mech. Appl. 401 (2014) 15–21.

[32] F. Xie, Z. Chen, J. Shang, et al., A Link prediction approach for item recommendation with complex number, in: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 1, IEEE, 2014, pp. 205–212.

[33] C. De Maio, G. Fenza, M. Gallo, et al., Social media marketing through time-aware collaborative filtering, in: Concurrency and Computation: Practice and Experience, 2017.

[34] A. Sara, Y.E.B. El Idrissi, R. Ajhoun, Time aware recommendation, in: Information and Communication Technology for The Muslim World, ICT4M, 2016, pp. 244–247.

[35] H. Liao, A. Zeng, R. Xiao, et al., Ranking reputation and quality in online rating systems, PLoS ONE 9 (5) (2014) e97146.

[36] Q.M. Zhang, A. Zeng, M.S. Shang, Extracting the information backbone in online system, PLoS ONE 8 (5) (2013) e62624.

[37] R. Rafeh, A. Bahrehmand, An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems, J. Inf. Sci. 38 (3) (2012) 205–221.