Knowledge-Based Systems 185 (2019) 104921

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Predictability of diffusion-based recommender systems[☆]

Peng Zhang [a], Leyang Xue [a], An Zeng [b,*]

[a] School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, PR China
[b] School of Systems Science, Beijing Normal University, Beijing 100875, PR China

ARTICLE INFO

Article history:
Received 17 December 2018
Received in revised form 1 August 2019
Accepted 3 August 2019
Available online 6 August 2019

Keywords:
Predictability
Diffusion-based algorithms
Recommender systems

ABSTRACT

Numerous diffusion-based recommendation algorithms (DBA) have been extended to improve the performance of such methods further. However, it is still not clear to what extent recommendation accuracy can be improved if we continue to extend existing algorithms. In this paper, we propose an ideal method to quantify the possible maximum recommendation accuracy of DBA, which is regarded as predictability of algorithms. Accordingly, the ideal method is applied to the extensively analyzed datasets. The result illustrates that the accuracy of DBA can still be improved by optimizing the resource allocation matrix on a dense network. Nevertheless, improving accuracy on sparse networks is difficult, mainly because the current accuracy of DBA is very close to its predictability. We find that the predictability can be enhanced effectively by multi-step resource diffusion, especially for inactive users (with less historical data). In contrast to common belief, there are plausible circumstances where the higher predictability of DBA does not correspond to active users. Additionally, we demonstrate that the recommendation accuracy is overestimated in the real online systems by random partition used in the literature, suggesting the recommendation in the real online systems may be a tough task.

1. Introduction

Due to the rapid development of information technology, there is a big gap between the ability to obtain useful information and the amount of data provided by the Internet. This situation makes it more difficult for users to select items that look similar to what they are looking for. Therefore, it is important to filter unrelated information and provide personalized recommendations to meet the multiple needs. Nowadays, recommender systems [1,2] have become a vital tool for online life. Many e-commerce websites rely on recommender systems to suggest consumers new possibly relevant items, e.g. Amazon, Netflix, YouTube, Alibaba, etc. [3,4].

The recommender system is a mature research field. Various recommendation algorithms have been proposed to solve the problem of information overload, including context-based analysis [5,6], collaborative filtering [7–9], latent semantic models [10,11], deep neural network [12,13] and so on [14,15]. While most recommendation algorithms are applied to data with rating, diffusion-based algorithms (DBA) can act on unary data without ratings [16–20] (See Ref [20] for a review of DBA), which has received considerable attention from a large number of mathematics and physics scholars. The DBA represents the input data with a user–item bipartite network where users link to items that they have bought [21]. Some well-known physical processes, such as random walk and heat conduction, are originally employed on the bipartite network to make a recommendation for users. Mass diffusion (MD) [17] and heat conduction (HC) [16] are regarded as one of the well-known recommendation algorithms in DBA. The hybrid of both MD and HC is proposed to solve the diversity–accuracy dilemma of recommender systems [18]. In addition, these algorithms have been extended in multiple directions by modifying the initial configuration of resource [22,23], updating the resource allocation ways [24–28], and personalizing the parameter incorporated into the algorithms [29–32].

In previous studies, some extensions are mainly used to improve the accuracy and diversity of recommendation. However, a fundamental problem is whether the recommendation accuracy can be improved effectively if we continue to extend the existing DBA on the bipartite network. In other words, it is not clear that what is the maximum accuracy of DBA that can be achieved. Similar problems are called predictability of algorithms by many scholars [33,34]. Current studies with predictability mainly include: the predictability of nine link prediction algorithms based on common neighbor similarity, in theory, is analyzed by Xu et al. [33]; the predictability of network is treated as an inherent property and can be reflected by structural consistency, which

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.104921.

* Corresponding author.
E-mail address: anzeng@bnu.edu.cn (A. Zeng).

https://doi.org/10.1016/j.knosys.2019.104921
0950-7051/© 2019 Elsevier B.V. All rights reserved.

has been proposed by Lü et al. [34]. Unlike these works above that consider the predictability of network as well as a specific link prediction algorithm, in this paper, we study the predictability of diffusion-based algorithms, which (i) guides us for developing better diffusion-based recommendation methods, (ii) deepens our understanding for the diffusion-based algorithms [35], and (iii) extracts the skeleton structure of the network to reduce computational complexity [36].

The new added value of this work is to (i) propose a method to quantify the predictability of DBA; (ii) show that existing DBA have great potential to improve recommendation accuracy on a dense network, however it is not effective to extend the DBA to further improve accuracy on a sparse network; (iii) find that predictability of DBA for inactive and active users (who select fewer and more objects) perform poorly and can be enhanced by multi-step resource diffusion; (iv) reveal the recommendation accuracy in real online systems is overestimated by random partition. The remainder of this paper is organized as follows. In Section 2, some recommendation algorithms based on diffusion are described in this paper. We introduce the ideal method proposed to quantify the predictability of DBA. In Section 3, we describe the datasets, dataset partition, and evaluation metrics. The analysis results are demonstrated in Section 4. We conclude this work with a brief outlook of future work in Section 5.

2. Methods

In this section, we propose an ideal method to quantify the predictability of DBA. Before introducing the method, we first describe the well-known mass diffusion (MD) [17] and the state-of-the-art similarity-preferential mass diffusion algorithm (SPMD) [30] (The comparison of accuracy among different algorithms can be seen in [20]). We clarify the resource diffusion process of various diffusion-based algorithms on the bipartite network. Finally, we introduce the ideal method and take an example to illustrate it.

2.1. Mass diffusion

Online commercial systems are naturally modeled as bipartite networks composed of two types of nodes, with the user set $U = \{U_1, U_2, \ldots, U_M\}$ and item set $O = \{O_1, O_2, \ldots, O_N\}$ (Note that an item denotes an online object that users can select). An edge denotes the relation between users and items, implying that a user has selected an item. The bipartite network can be represented by a $M \times N$ matrix A (M users and N items) where the element $A_{i\alpha} = 1$ if a user i has selected an item α and $A_{i\alpha} = 0$ otherwise.

The mass diffusion (MD) [17] is described as follows: For a given user i, we initially set the unit amount of resources on those items that user i has selected, and other items have zero initial resource value. The resource of each item is then evenly distributed to their neighbor users who have collected this item, which means that users receive the resource value $1/k_\alpha$ from the item α. The resource value on users is again divided evenly and spread over the network from the user side back to the item side. The final resource value of items represents the recommendation score by which we can generate the recommendation list for users. A mathematic formulation of the mass diffusion is as follow:

$$f'^{i}_{\alpha} = \sum_{\beta} W_{\alpha\beta} f^{i}_{\beta} = (Wf^{i})_{\alpha}, \tag{1}$$

where f^i denotes the initial resource vector for user i and its elements as $f^i_\beta = a_{i\beta}$. f'^{i}_{α} is the final resource value of item α for user i. The $W_{\alpha\beta}$ is an element of resource allocation matrix W and described from the following equation of probabilistic spreading:

$$W_{\alpha\beta} = \frac{1}{k_\beta} \sum_i \frac{a_{i\alpha} a_{i\beta}}{k_i}, \tag{2}$$

where k_i denotes the number of items that user i has selected, namely the degree of user i node. The k_β refers to the number of users that have chosen this item β, called the degree of item β node. $W_{\alpha\beta}$ means the distribution weight of resource from item β to α by two-step diffusion. The matrix W, the resource allocation matrix, represents the item-side projection of a bipartite network. Note that this matrix is asymmetric, $W_{\alpha\beta} \neq W_{\beta\alpha}$.

2.2. Similarity preference mass diffusion

The similarity preferential mass diffusion (SPMD) [30] is designed from the perspective of user similarity. An additional model parameter is incorporated into the mass diffusion, which is used to increase the influence of users who are most similar to a target user. The description of SPMD recommendation algorithm is as follows: For a given user i, we firstly assign each item collected by the target user with a unit resource. Then, the resource of each item is evenly distributed to its neighbor users. User j receives the resource from the item α that user i has collected. The final total resource of user j receiving from the user i can be represented as

$$f^{SPMD}_{ij} = \sum_{\alpha=1}^{N} \frac{a_{i\alpha} a_{j\alpha}}{k_\alpha}, \tag{3}$$

where f_{ij} indicates the similarity between user i and user j. In order to boost the influence of similar users, the similarity value is modified from f_{ij} to f^{θ}_{ij}. Finally, the corrected resource value of user j is again evenly distributed its neighbor item β as before. The resource value of item β is obtained by summing over all similar users. Mathematical formula of the SPMD is

$$f^{SPMD}_{i\beta} = \sum_{j=1}^{M} \frac{a_{j\beta} f^{\theta}_{ij}}{k_j}. \tag{4}$$

The recommendation list of user i is obtained by sorting the resource value of all unselected items that correspond to $f_{i\beta}$. Here, the parameter θ is tunable. The result of recommendation degenerates to original mass diffusion when $\theta = 1$, enhances the weight of similar users when $\theta > 1$, and suppresses the weight of similar users when $\theta < 1$ (Note that user weight decreases with similarity when $\theta < 0$, which is not reasonable). By tunning the parameter θ, we can get the optimal recommendation result.

2.3. Resource diffusion process

In addition to two recommendation algorithms described above, there are various algorithms based on diffusion. Although the essential difference among DBA is the configuration of the initial resource, the way of allocating resource, and the method to calculate the final resource for each item, there still have a common feature: the resource diffusion process. Personalized recommendations that rely on resource diffusion mainly includes the following three steps [1]:

- The initial resource of items is acquired by the target user's historical preferences on a given bipartite network.
- Each item distributes the initial resource to all users who have collected it according to a way of resource allocation
- Each user reallocates the resource to all items collected by the user according to a method of resource allocation.

When the resource goes from one type of nodes to the other type of nodes, it is called a step of diffusion. Through the above three steps, the recommendation can be made. In fact, a recommendation can also be based on diffusion with more than three steps. For instance, after five steps of diffusion, the resource reaches again the item side, with the ranking of which the recommendation list can be generated.

2.4. Ideal method

Predictability is usually defined as the possible maximum accuracy of a recommendation algorithm [34,37]. Intuitively, the predictability of recommendation algorithms should be 1 with such definition because those unselected items are always distinguishable. However, the DBA has fundamental limits on the recommendation of items. For instance, some items cannot receive the resource from the target users at a certain number of diffusion steps, making these items unpredictable. Therefore, the amount of items covered by the resource is determined by the number of diffusion steps or intrinsic network structures. Inspired by this idea, we propose an ideal method to quantify the upper bound of recommendation accuracy of DBA with a given number of diffusion steps. The ideal method is described as follows: Firstly, we assume that there is an ideal resource allocation way that will always give the diffusion-reachable probe set items (this refer to items covered by the resource in the probe set) with the highest recommendation scores. Moreover, those unselected items for a target user are sorted to generate the recommendation list according to the recommendation scores (the resource value of items), which is equal to put diffusion-reachable probe set items into the front of the recommendation list. Finally, the theoretical upper bound of recommendation accuracy is obtained by measuring the recommendation list. Specifically, we adopt the ideal method, a virtual method meaningful for its guidance on developing better DBA, to understand the upper limit of DBA's accuracy. In the simulation, the ideal method is implemented by artificially putting those diffusion-reachable probe set items into the front of the recommendation list. Then, we employ ranking score [17] (the smaller the better, see definition in Section 3.3) to measure the recommendation accuracy as predictability of this kind of algorithms.

A simple example of this procedure is illustrated in Fig. 1. The original data is divided into a training set (E_T) and a probe set (E_P). A training set is viewed as the historical information of users. A probe set is considered as a set of items that will be brought by users. In the dataset, the target user (baby blue) has selected the $item_1$, $item_3$, and will buy the $item_2$, $item_4$ and $item_5$. We respectively use the MD and ideal method to recommend items for the target user. The recommendation result of MD is 6, 4, 5, 2, while the ideal method is 4, 5, 6, 2. The ideal method is implemented by putting $item_4$ and $item_5$ into the front of the recommendation list, while the position of $item_2$ in the recommendation list is not changed because the initial resource of target users cannot cover it by three-step diffusion. The three-step diffusion of resource is equivalent to the three-hop random walk began from the target user on the bipartite network. Those items unreached by diffusion are unpredictable, which cannot be solved by optimizing the weight matrix of resource diffusion. Finally, we measure the prediction accuracy of recommendation algorithms by ranking score. The ranking score of MD and ideal method are 0.75 and 0.583 respectively. Consequently, the predictability of DBA is 0.583.

3. Experiment

We conduct several experiments to analyze the result of predictability. Seven datasets are chosen to conduct experiments: Movielens, Netflix, Amazon, Delicious, Douban, Epinions, and Stack. We select the ranking score as the evaluation metric of recommendation accuracy.

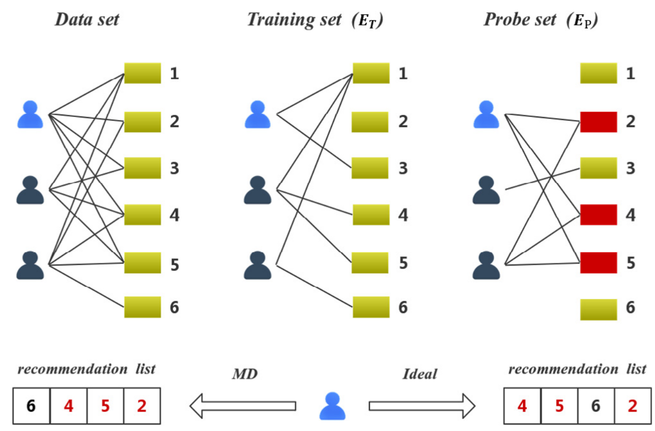

Fig. 1. A simple illustrative example of ideal method. The user in blue is the target user, and rectangles denote items. The red rectangle denotes objects in the probe set that will be bought by a target user. Recommendation lists generated by MD and ideal method are shown, where the recommendation list of the ideal method is generated by top ranking the probe-set objects whose resource can be reached by the three-step diffusion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1. Datasets

In the simulation, we use two benchmark datasets and five subsets randomly selected from the original dataset: Movielens,[1] Netflix,[2] Epinions,[3] Stack,[4] Amazon,[5] Delicious[6] and Douban.[7] We randomly extract the subset from the original dataset because the original dataset is too big to be processed. That the size of the subset is determined relies on the memory storage capacity. Due to the random extraction, the size of the subset does not affect the qualitative outcome of the experiment. Movielens is a web-based personalized recommendation system to recommend movies for users. The Movielens data consists of 100000 ratings (1–5) resulting from 943 users remarked 1682 movies and contains the seven months from September 19th, 1997 to April 22nd, 1998. The Netflix dataset is a subset of the original dataset released for the Netflix Prize. The data contains 1014 users and 1977 movies chosen from the original data at random and all links among them. 62848 links and time stamp information are contained in the subset network. Epinions is an online product rating site. Users and products consist the Epinions dataset. Each edge connects a user with a product and represents a rating. Stack Overflow is the main question and answer website of the Stack Exchange Network where nodes represent users and posts and an edge denotes that a user has marked a post as a favorite. Amazon, Delicious and Epinions dataset are available by crawling their website. The above six subsets are randomly selected from the original dataset. Table 1 demonstrates the detailed statistical characteristics of the seven networks. The Netflix network is sparser than the Movielens. Movielens and Netflix are used to analyze the Figs. 2 to 6. Fig. 7 is based on all datasets.

1 https://grouplens.org/datasets/movielens/.
2 http://konect.uni-koblenz.de/networks/netflix.
3 http://konect.uni-koblenz.de/networks/epinions.
4 http://konect.uni-koblenz.de/networks/stackexchange-stackoverflow.
5 https://www.amazon.com/.
6 http://www.thedeliciousgroup.com/.
7 https://www.douban.com/.

Table 1
The detailed statistical characteristics of the seven experimental datasets.

Datasets	Users	Items	Links	$\langle k_{user} \rangle$	$\langle k_{item} \rangle$	Sparsity
Movielens	943	1,682	100,000	106.04	59.45	6.30×10^{-2}
Netflix	1,014	1,977	62,848	61.98	31.79	3.13×10^{-2}
Amazon	8,071	73,071	101,155	12.53	1.38	0.02×10^{-2}
Delicious	1,000	77,371	129,231	129.23	1.67	0.17×10^{-2}
Douban	1,699	59,102	195,372	114.99	3.31	0.19×10^{-2}
Epinions	1,205	106,685	155,121	128.73	1.45	0.12×10^{-2}
Stack	54,520	38,904	131,921	2.42	3.39	0.06×10^{-3}

Table 2
Result (*RS*) achieved by different recommendation algorithms. Parameter values (if any) were always set by maximizing the ranking score for each individual method. The optimal parameter θ^* of SPMD is $\theta = 0.26, 0.03, 0.19, 0.01$ for the Movielens (T:0.9 P:0.1), Movielens (T:0.1 P:0.9), Netflix (T:0.9 P:0.1) and Netflix (T:0.1 P:0.9), respectively.

Algorithm	Dataset			
	Movielens T:0.9 P:0.1	Movielens T:0.1 P:0.9	Netflix T:0.9 P:0.1	Netflix T:0.1 P:0.9
MD	0.0837	0.2478	0.0573	0.2187
SPMD	0.0750	0.2449	0.0553	0.2166
Ideal	0.0034	0.1660	0.0023	0.1904

3.2. Datasets partition

In order to analyze the result, we use two ways to divide datasets: random partition and temporal division. (1) Random division refers to that all links in the bipartite network are randomly divided into a training set (E_T) and a probe set (E_P) according to the partition proportion. (2) Temporal partition is based on timestamps on links (i.e. earlier links are put in the training set and later links are put in the probe set). The training set is viewed as known information to generate the recommendation. The probe set is used to quantify the prediction accuracy. Obviously, $E_T \bigcap E_P = \emptyset$ and $E_T \bigcup E_P = E$. All the simulation results are obtained by averaging over fifty independent experiments when the dataset is divided randomly into a training set and probe set.

3.3. Evaluation metrics

We employ the ranking score to quantify the maximum recommendation accuracy of DBA. In addition, we propose a metric called diffusion coverage to quantify the range of resource diffusion.

(1) Ranking Score (*RS*) [17]: The ranking score measures the prediction accuracy of a recommendation algorithm which generates an ordered queen of all uncollected items for an arbitrary user to match the user's preference in the future. For a target user u_i, the recommendation list is generated by a recommendation algorithm. The edge $u_i - o_\alpha$ in probe set, we measure the position of object o_α in recommendation list of user u_i. A good recommendation algorithm is expected to give the object in the probe set a higher rank, which leads to a small ranking score. The ranking score of a target user is obtained by averaging over all entries in the probe set. The specific formula is as follows:

$$RS_{u_i} = \frac{1}{|\{u_{i\alpha} \in E^P\}|} \sum_{u_{i\alpha} \in E^P} \frac{l_{i\alpha}}{L_{u_i}}, \tag{5}$$

where RS_{u_i} denotes the ranking score of the user *i*. The $u_{i\alpha}$ is user *i* - object α relations in the probe set and $l_{i\alpha}$ is the rank of object α in the recommendation list of user *i*. L_{u_i} equals to $|O - E^T_{u_i}|$, namely the number of uncollected items for user *i*. $|\{u_{i\alpha} \in E^P\}|$ denotes the number of items for user *i* in probe set. The *RS* of the whole system is obtained by averaging RS_{ui} over all users. Obviously, the range of *RS* is that $RS \in [0, 1]$. The smaller *RS*, the higher the prediction accuracy of recommendation algorithms, vice versa. We use the ranking score to measure recommendation accuracy. On one hand, we do not need to set the length of the recommendation list (L) because *RS* takes into account all items in the probe set. For the precision and recall, researchers usually set the L as 5, 10 or 20 to calculate the recommendation accuracy. In fact, the value of L has a significant effect on the predictability of DBA. The predictability (accuracy upper bound) is 1 if the L is smaller than the number of diffusion-reachable items in the probe set. Otherwise, the predictability varies over the value of L. On the other hand, *RS* can be used to examine the rank of all items that users like. *RS* is thus a more precise metric as it considers more complete information.

(2) Diffusion Coverage (*DC*): Diffusion coverage quantifies the fraction of uncollected items that resource can cover by DBA. *DC* is defined as the ratio of unselected items covered by resource as compared to the total number of uncollected items. For a target user, the larger the range of resource diffusion, the more items can be possibly recommended to users in this system. The definition is as follows:

$$DC_{u_i} = \frac{n_{u_i}}{N_{u_i}}, \tag{6}$$

where n_{u_i} denotes the number of uncollected items that the resource of target user *i* can cover. N_{u_i} is the total number of uncollected items. Obviously, $DC_{u_i} \in [0, 1]$. The larger the *DC*, the greater the resource diffusion coverage.

4. Results

In this section, we compare the accuracy between DBA and predictability. Then we analyze a serious of results about the predictability of different users. Further, we propose a method to improve the predictability. In addition, we also investigate the predictability with the temporal partition.

4.1. The predictability of diffusion-based algorithms

According to a recent review article, we find that similarity-preferential mass diffusion (SPMD) is one of the state-of-the-art diffusion-based algorithms [20]. We employ the SPMD and MD to conduct the experiment and compare their accuracy with the ideal case. The result achieved by three methods can be seen in Table 2. The recommendation accuracy of the SPMD is better but still close to the mass diffusion in different datasets. Its predictability (*RS* of ideal method) shows that the accuracy of diffusion based algorithms (DBA) can still be improved.

The analysis of predictability for different users on dense and spares network reveals three general results (see Fig. 2): (1) For users with different degrees, the *RS* of ideal method is close to 0 on a dense network. This means that items can be accurately recommended by DBA if we continue to update the weight matrix of resource diffusion. In other words, the current recommendation accuracy of DBA can be enhanced if the scoring scheme of the diffusion-reachable items is improved (see Fig. 2a, b). (2) The *RS* of ideal method for both inactive and active users is larger than other users on the sparse network, where the inactive/active refer to users whose degree is smaller than 3/larger than 54, suggesting that the predictability for these users is low and it is not easy to recommend exactly what these users like on a sparse network (see Fig. 2c, d). (3) The *RS* of MD for various users is close to the ideal method on the Netflix dataset (see Fig. 2d). In this case, if we are dedicated to extending existing DBA, the improvement

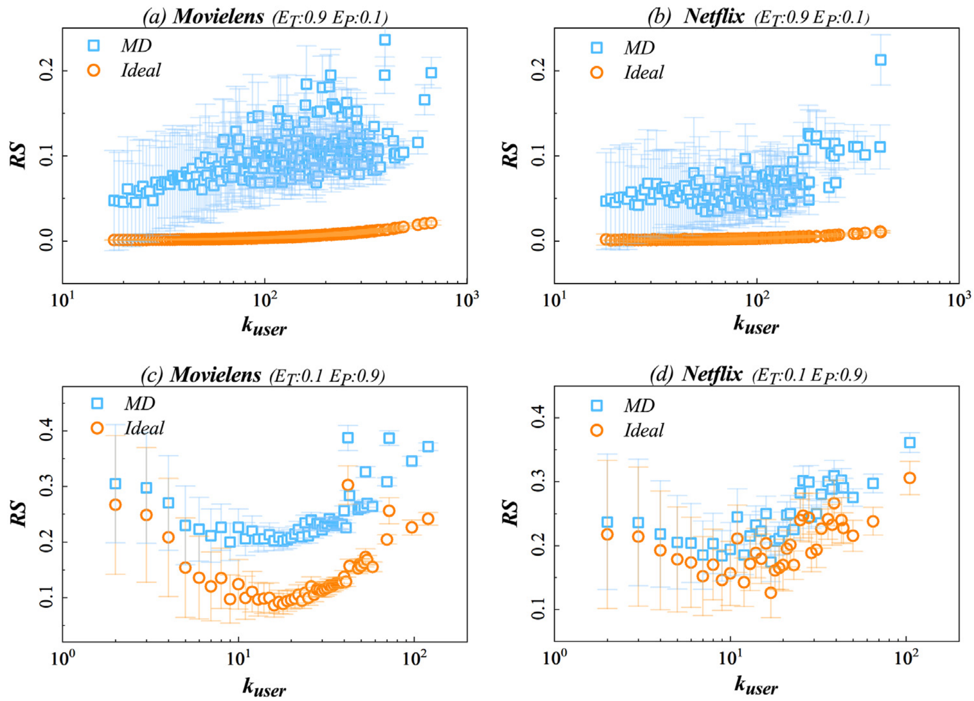

Fig. 2. The predictability of DBA (*RS* of ideal method) and recommendation accuracy of mass diffusion (MD *RS*) for users with different degrees. Points are averages over ranking score with the same degree of user. a, b The Movielens and Netflix datasets are divided into E_T and E_P respectively according to the ratio of 90% to 10%, making a recommendation on a dense network. c, d Dividing the datasets into E_T and E_P is based on the ratio of 10% to 90%, conducting experiment on a sparse network.

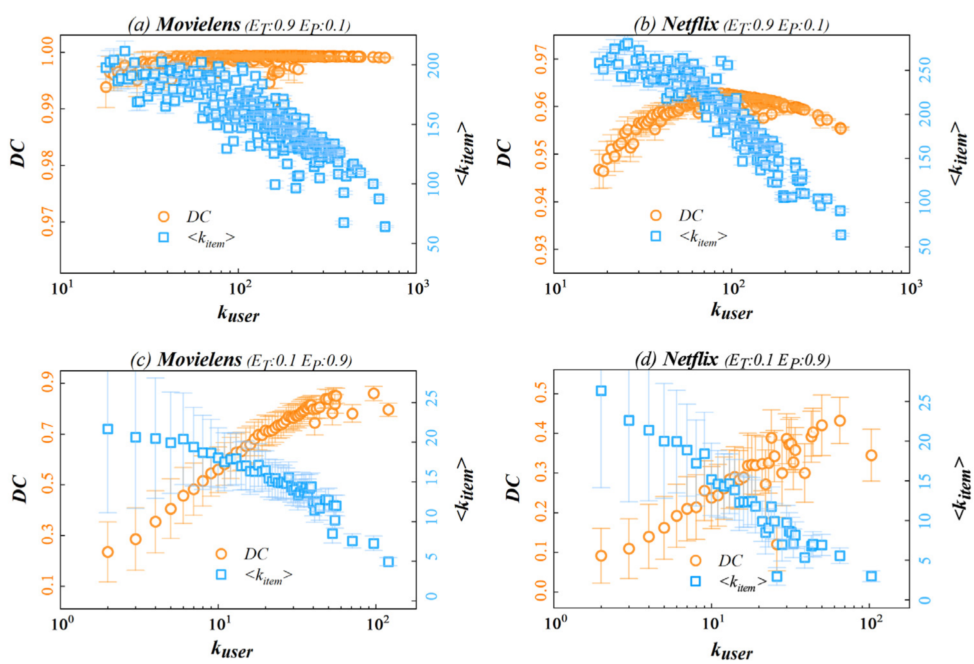

Fig. 3. The diffusion coverage (*DC*) and average degree of item $\langle k_{item} \rangle$ for users with different degrees. The data division setting as well as the averaging procedure are the same as those in Fig. 2. a, c Movielens. b, d Netflix.

of recommendation accuracy is not obvious. Enhancing the recommendation accuracy of DBA on a sparse network needs to take other methods to increase its predictability instead of simply adjusting the resource allocation matrix. Our study highlights the fact that whether a new recommendation algorithm based on resource diffusion would be proposed relies on its predictability.

Intuitively, the low predictability of DBA for inactive users is not surprising in a sparse network, because the small amount of link information makes it difficult to diffuse resource. However, for active users, the low predictability of DBA is contrary to intuition. We propose two metrics illustrate the mechanism behind it: diffusion coverage (*DC*) and average degree of items that a user selects $\langle k_{item} \rangle$ (detailed information of diffusion coverage can be seen in Eq. (6)). The $\langle k_{item} \rangle$ reflects the popularity of items selected by users. The *DC* and $\langle k_{item} \rangle$ as a function of k_{user} on dense and sparse network is shown in Fig. 3. (1) On a dense

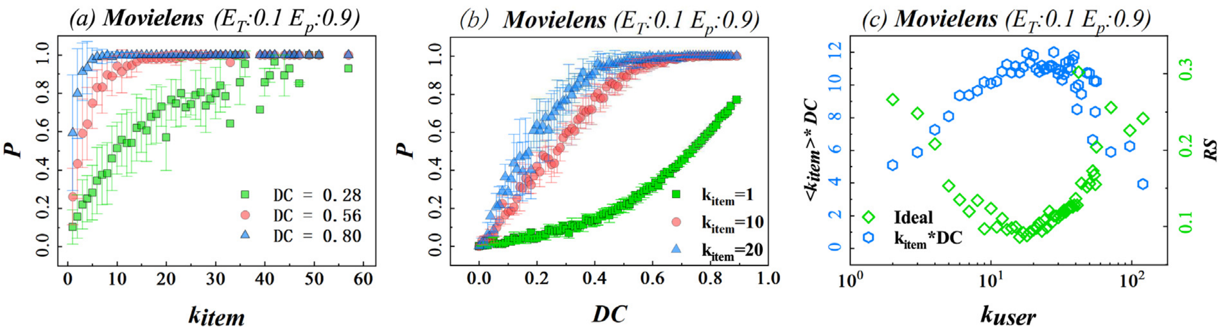

Fig. 4. a. P plotted as a function of k_{item}, showing a positive correlation between P and the k_{item} with the given value of DC, where the P represents the ratio of items covered by resource in the probe set. The k_{item} denotes the degree of the item node. b. P plotted as a function of DC, showing the positive correlation between P and DC when the k_{item} is fixed. c. $DC * \langle k_{item} \rangle$ and RS plotted as a function of k_{user}, which is used to confirm that the predictability of DBA is determined by the DC and k_{item}.

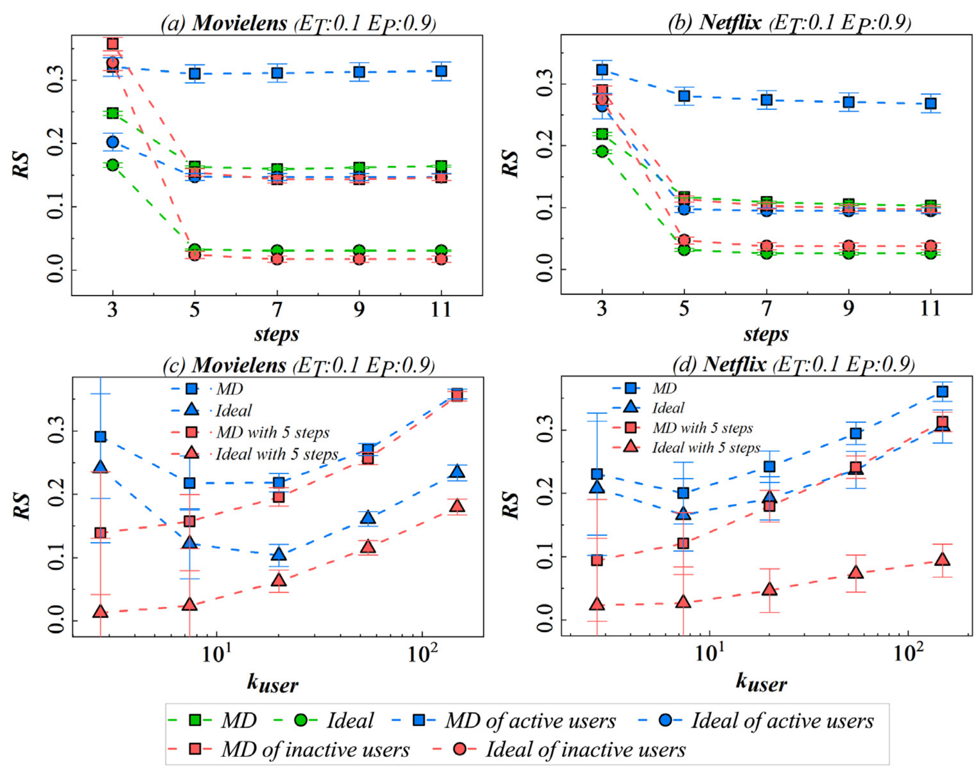

Fig. 5. The predictability of DBA (RS of ideal method) and ranking score of mass diffusion (MD RS) by muti-step resource diffusion. a, b The ranking score of MD and ideal method for different diffusion steps on a sparse network. Here, inactive /active users refer to user's degree less than 3/greater than 54 respectively. c, d Scatter plot of the ranking score as a function of user degree, which is conducted on Movielens and Netflix datasets respectively. The ranking score of MD and ideal method with three steps and five steps are shown on a sparse network. Note that when the number of diffusion step is three, the method degenerates to mass diffusion [17]. Points are averages over ranking score binned in intervals of the exponent.

network (see Fig. 3a), the smallest DC among all users is 0.99, suggesting that the resource for most of unselected items can be reached by diffusion. The $\langle k_{item} \rangle$ for all users is very large even if it decreases with k_{user}. The result shows that users tend to select popular items that are covered easily by resource, which is a possible explanation for the higher predictability of DBA observed on the dense network. (2) On a sparse network (see Fig. 3c), the DC for inactive users is very small, reflecting the fact that small degree users propagate initial resource to less unselected items, which leads to the lower predictability of DBA. Regarding active users, although DC is larger than the inactive, it cannot reach 1. In this situation, $\langle k_{item} \rangle$ also influences the predictability. The smaller $\langle k_{item} \rangle$ suggests that active users select many cold items that can be seldom reached by diffusion, resulting in the lower predictability. (3) Indeed, the predictability is best when the two curves are close to each other(see Figs. 3c and 2c). Thus, we propose a metric P, the ratio of items in the probe set that can be reached by diffusion, to explain the interesting phenomenon. One could see that the P increases with k_{item} given a certain DC (see Fig. 4a). Similarly, P also increases with the DC given a certain k_{item} (see Fig. 4b). As both DC and k_{item} influence P, we can write their relation as $P \sim DC * k_{item}$. The product of k_{item} and DC cannot be large if only one of the factor is large while the other is small. When both factors are relatively large, the product achieves maximum. As k_{item} monotonously decreases with k_{user} while DC monotonously increases with k_{user}, there is eventually an optimal value of k_{user} for P, resulting an optimal predictability (see Fig. 4c). Similar interpretation on Netflix dataset is obtained from Fig. 3b, d. The proposition of two metrics deepens us understanding for limitations and advantage of DBA.

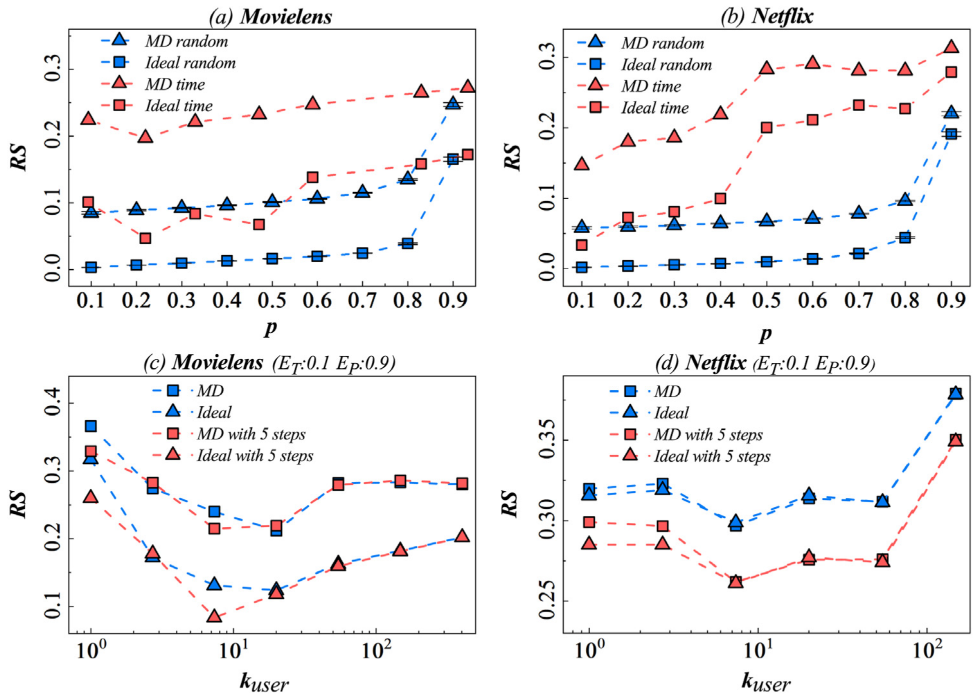

Fig. 6. The predictability of DBA (*RS* of ideal method) and ranking score of mass diffusion (MD *RS*) according to temporal partition. a, b The ranking score plotted as a function of *p*, where *p* represents the ratio of the probe set. Black bars is the standard error. c, d Scatter plot of the ranking score as a function of user degree, which is conducted on Movielens and Netflix datasets respectively. The ranking score of MD and ideal method with three steps and five steps are also shown on a sparse network. The averaging procedure is the same as those in Fig. 5 c, d.

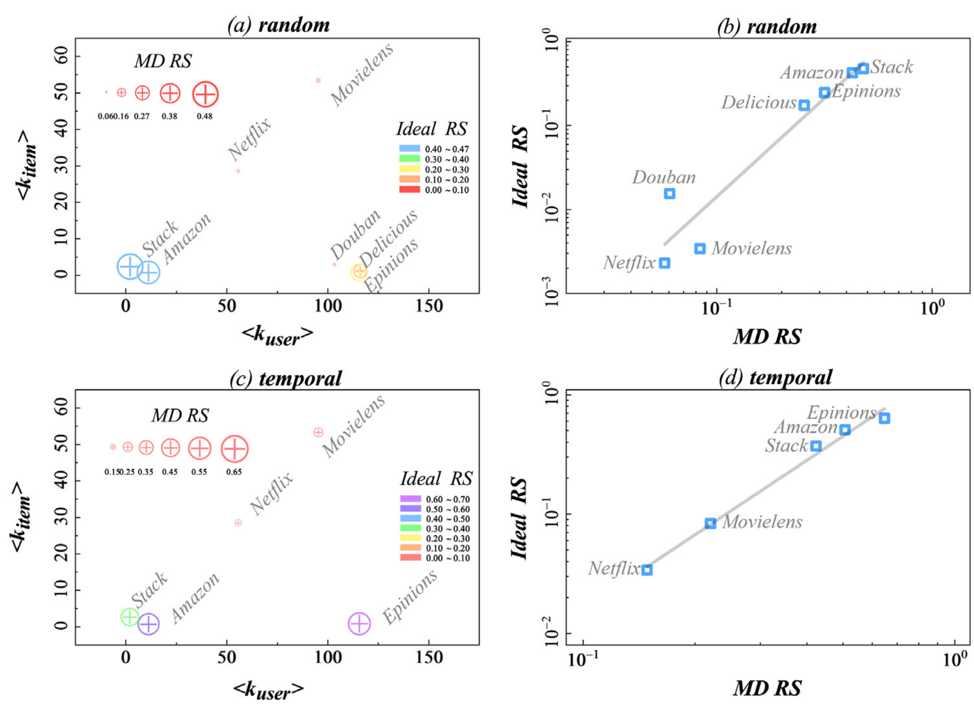

Fig. 7. The predictability of DBA (*RS* of ideal method) and ranking score of mass diffusion (MD *RS*) on different datasets. a, c The location of different datasets in the space of average degree on item and user, with the size and color of each point as the ranking score of MD and ideal method. b, d The ranking score of ideal method versus the ranking score of the MD in different datasets. a, b show the results with random division while c, d show the results with temporal division. A fitted line is plotted in each figure to guide the eyes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. The improvement of predictability of diffusion-based methods

The lower recommendation accuracy of MD observed from Fig. 2c, d, is mainly constrained by its predictability. We next turn our attention to improve the predictability of DBA. Inspired by diffusion coverage, we enhance the predictability on the sparse network by multi-step resource diffusion. The *RS* of MD and ideal method with multi-step diffusion is shown in

Table 3
The ranking score of MD and ideal method on the seven experiment datasets with random and temporal partition.

Partition way	Method	Datasets						
		Netflix	Movielens	Amazon	Epinions	Stack	Delicious	Douban
Random	MD	0.0573	0.0837	0.4261	0.3168	0.4813	0.2554	0.0605
	Ideal	0.0023	0.0034	0.4243	0.2471	0.4739	0.1742	0.0155
Temporal	MD	0.1487	0.2203	0.5069	0.6472	0.4238	–	–
	Ideal	0.0039	0.0834	0.5058	0.6317	0.3701	–	–

Fig. 5. One could see that the *RS* of MD and ideal method with five-step diffusion for different users reaches the smallest, which shows that the predictability can be improved largely, especially for inactive users (see Fig. 5a, b). This further confirms our explanation that low predictability for inactive users is caused by the limited diffusion in Fig. 3c. In Fig. 5c, d, we also compare the difference of accuracy between three-step and five-step diffusion. The result indicates that the accuracy of MD for inactive users can be improved largely on the sparse network. Actually, this is mainly because the predictability of DBA is enhanced by five-step diffusion.

4.3. The predictability of diffusion-based algorithms with temporal partition

In the recommendation, it is more important to acquire the temporal information of items. For example, the taste of users changes over time. Accordingly, we analyze the predictability of DBA according to the temporal partition, and compare recommendation accuracy with different partition ways. We plot the *RS* as a function of p for both MD and ideal method with random and temporal partition in Fig. 6a, b, where p refers to the proportion of probe set. Obviously, the recommendation accuracy of MD and predictability with temporal partition is much lower than random division, suggesting the accuracy of recommendation in the real online systems is overestimated by random partition. In addition, the difference in accuracy between MD and predictability gradually decreases with partition proportion p, a sparser dataset (Netflix) in particular. This reveals that we should seek other ways to increase its predictability if we want to improve the accuracy of DBA on a sparse network. To observe the predictability for various users under the temporal partition, the *RS* of both MD and ideal method with three-step and five-step diffusion is shown in Fig. 6c, d, our finding is not specific to the result in Fig. 5c, d, with the same conclusion.

In fact, the density of the link in the network directly determines the prediction ability of DBA. To better understand how network sparsity affects the predictability, two metrics are used to characterize different networks: the average degree of user $\langle k_{user} \rangle$ and the average degree of item $\langle k_{item} \rangle$. The average degree of user denotes the average number of items selected by users, suggesting the likelihood that a user will buy items. The average degree of items is the average number of items selected, indicating the likelihood that users will select the item. Fig. 7a, c show *RS* of both ideal method and MD for different kinds of datasets when datasets are embedded in the space of average degree on item and user. We find that the greater the average degree of users and items, the higher the predictability of DBA will be demonstrated. However, for those networks that only have a higher average degree of users, the predictability of DBA is not clear. For instance, the predictability in the Douban dataset is very high with random division, while the predictability in Epinions is relatively low. Moreover, the predictability with random partition is higher than that of temporal division in all datasets, except for Stack. The point of Delicious is covered by Epinions (see Fig. 7a, more detailed information can be seen in Table 3). Finally, we verify the effectiveness of the ideal method under different sparsity of datasets. The scatter plot of *RS* of ideal method as a function of MD *RS* in all datasets is shown in Fig. 7b, d. It is not surprising that the RS of MD is positively correlated with the ideal method, which suggests the recommendation accuracy of MD is constrained by its predictability.

5. Conclusions

Although many studies have focused on improving the accuracy of diffusion-based algorithms, the extent to which network can be predicted by DBA lacks of understanding. In this paper, we mainly study the maximum predictive ability of diffusion-based algorithms, or called the predictability of DBA. We propose a method to quantify its predictability, which allows to evaluate existing diffusion-based algorithms and provides a guideline for designing some new resource diffusion methods. Through comparing the accuracy between ideal method and MD, we find that the predictability is very high on a dense network and unselected items can be accurately recommended if we continue to optimize the resource allocation matrix. In contrast, on a sparse network, the recommendation accuracy of MD is very close to its predictability. In this case, it is ineffective to extend the existing diffusion-based algorithms to improve accuracy. Interestingly, the predictability for the inactive and active users is low on a sparse network. We propose diffusion coverage and item average degree to illustrate the mechanism behind it. The key insight of our finding is that active users tend to select unpopular items that are hardly covered by three-step diffusion, and the inactive users propagate the initial resource to fewer items. Meanwhile, we show that the predictability could be improved profoundly by multi-step resource diffusion, especially for inactive users. We also find the predictability of DBA with the temporal dataset partition is lower than that of random partition, revealing that recommendation accuracy in the real online system is overestimated by random partition. Our results obtained from the ideal method can be applied to many recommendation algorithms considering the structural information of user–item bipartite network. Take collaborative filtering as an example, the predictability estimation can work if the similarity of users or items is calculated by metrics based on local network information, such as the common neighbor, Jaccard similarity and so on.

The predictability of DBA allows us to evaluate various recommendation algorithms based on resource diffusion. This is instructive for proposing some new diffusion-based algorithms. For example, we could focus on enhancing the recommendation accuracy when the accuracy of algorithms are far from predictability because these algorithms have great potential to improve. However, for those algorithms whose accuracy is close to the predictability, we should find other methods to improve the predictability. A multi-step diffusion method is presented to improve the predictability of DBA on a sparse network. In addition, there are still many other methods to achieve higher predictability, such as coupling social networks of the user or adding virtual edges. Moreover, the predictability of DBA can also extract the network skeleton structure [36] and mine key links. We can remove those edges that cannot affect the predictability of recommendation algorithms, so that the recommendation algorithms only need to process a small amount of data, with the effect of reduction on computational complexity accordingly. In our research, we only explore the predictability of DBA, while it is unknown how network structure affects the predictability of DBA. This is an interesting question open for future research.

CRediT authorship contribution statement

Peng Zhang: Formal analysis, Supervision, Writing - review & editing. **Leyang Xue:** Formal analysis, Software, Visualization, Writing - original draft. **An Zeng:** Conceptualization, Formal analysis, Supervision, Writing - review & editing, Methodology.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61403037, No. 61603046), the Natural Science Foundation of Beijing, China (Grant No. L160008).

References

[1] L. Lü, M. Medo, C.H. Yeung, Y.-C. Zhang, Z.-K. Zhang, T. Zhou, Recommender systems, Phys. Rep. 519 (1) (2012) 1–49, recommender Systems.
[2] J. Protasiewicz, W. Pedrycz, M. Kozłowski, S. Dadas, T. Stanisławek, A. Kopacz, M. Gałężewska, A recommender system of reviewers and experts in reviewing problems, Knowl.-Based Syst. 106 (2016) 164–178.
[3] E.R. Núñez-Valdez, D. Quintana, R.G. Crespo, P. Isasi, E. Herrera-Viedma, A recommender system based on implicit feedback for selective dissemination of ebooks, Inform. Sci. 467 (2018) 87–98.
[4] A. Zeng, C.H. Yeung, M. Medo, Y.-C. Zhang, Modeling mutual feedback between users and recommender systems, J. Stat. Mech. Theory Exp. 2015 (7) (2015) P07020.
[5] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A.C. Pierewan, H. Chiroma, T. Herawan, Context-aware recommender system: A review of recent developmental process and future research direction, Appl. Sci. 7 (12) (2017).
[6] N.M. Villegas, C. Sánchez, J. Díaz-Cely, G. Tamura, Characterizing context-aware recommender systems: A systematic literature review, Knowl.-Based Syst. 140 (2018) 173–200.
[7] M.D. Ekstrand, J.T. Riedl, J.A. Konstan, Collaborative filtering recommender systems, Found. Trends Hum. Comput. Interact. 4 (2) (2011) 81–173.
[8] M. Srifi, B.A. Hammou, S. Mouline, A.A. Lahcen, Collaborative recommender systems based on user-generated reviews: A concise survey, in: 2018 International Symposium on Advanced Electrical and Communication Technologies, ISAECT, 2018, pp. 1–6.
[9] H. Zhang, X. Zhang, Z. Tian, Z. Li, J. Yu, F. Li, Incorporating temporal dynamics into lda for one-class collaborative filtering, Knowl.-Based Syst. 150 (2018) 49–56.
[10] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 89–115.
[11] R. Wang, H.K. Cheng, Y. Jiang, J. Lou, A novel matrix factorization model for recommendation with LOD-based semantic similarity measure, Expert Syst. Appl. 123 (2019) 70–81.
[12] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (1) (2019) 5:1–5:38.
[13] Z. Batmaz, A. Yurekli, A. Bilge, C. Kaleli, A review on deep learning for recommender systems: challenges and remedies, Artif. Intell. Rev. 52 (1) (2019) 1–37.
[14] C. Xu, A big-data oriented recommendation method based on multi-objective optimization, Knowl.-Based Syst. 177 (2019) 11–21.
[15] Y. Qian, Y. Zhang, X. Ma, H. Yu, L. Peng, Ears: Emotion-aware recommender system based on hybrid information fusion, Inf. Fusion 46 (2019) 141–146.
[16] Y.-C. Zhang, M. Blattner, Y.-K. Yu, Heat conduction process on community networks as a recommendation model, Phys. Rev. Lett. 99 (2007) 154301.
[17] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, Bipartite network projection and personal recommendation, Phys. Rev. E 76 (2007) 046115.
[18] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J.R. Wakeling, Y.-C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. 107 (10) (2010) 4511–4515.
[19] M. Medo, Y.-C. Zhang, T. Zhou, Adaptive model for recommendation of news, Europhys. Lett. 88 (3) (2009) 38005.
[20] F. Yu, A. Zeng, S. Gillard, M. Medo, Network-based recommendation algorithms: A review, Physica A 452 (2016) 192–208.
[21] M.-S. Shang, L. Lü, Y.-C. Zhang, T. Zhou, Empirical analysis of web-based user-object bipartite networks, Europhys. Lett 90 (4) (2010) 48006.
[22] T. Zhou, L.-L. Jiang, R.-Q. Su, Y.-C. Zhang, Effect of initial configuration on network-based recommendation, Europhys. Lett. 81 (5) (2008) 58004.
[23] C. Liu, W.-X. Zhou, Heterogeneity in initial resource configurations improves a network-based hybrid recommendation algorithm, Physica A 391 (22) (2012) 5704–5711.
[24] T. Zhou, R.-Q. Su, R.-R. Liu, L.-L. Jiang, B.-H. Wang, Y.-C. Zhang, Accurate and diverse recommendations via eliminating redundant correlations, New J. Phys. 11 (12) (2009) 123008.
[25] G. Chen, T. Gao, X. Zhu, H. Tian, Z. Yang, Personalized recommendation based on preferential bidirectional mass diffusion, Physica A 469 (2017) 397–404.
[26] T. Gao, Y. Zhang, X. Zhu, L. Li, Personalized Recommendation based on unbalanced symmetrical mass diffusion, in: 2017 IEEE Third International Conference on Multimedia Big Data, BigMM, 2017, pp. 384–388.
[27] X. Zhu, H. Tian, P. Zhang, Z. Hu, T. Zhou, Personalized recommendation based on unbiased consistence, Europhys. Lett. 111 (4) (2015) 48007.
[28] X. Zhu, H. Tian, S. Cai, Personalized recommendation with corrected similarity, J. Stat. Mech. Theory Exp. 2014 (7) (2014) P07004.
[29] G. Qiang, S. Wenjun1, H. Zhaolong, H. Lei, Z. Yilu1, C. Fangjiao, Non-equilibrium mass diffusion recommendation algorithm based on popularity, J. Comput. Appl. 35 (12) (2015) 3502.
[30] A. Zeng, A. Vidmer, M. Medo, Y.-C. Zhang, Information filtering by similarity-preferential diffusion processes, Europhys. Lett. 105 (5) (2014) 58002.
[31] D.-C. Nie, Y.-H. An, Q. Dong, Y. Fu, T. Zhou, Information filtering via balanced diffusion on bipartite networks, Physica A 421 (2015) 44–53.
[32] R.-R. Liu, J.-G. Liu, C.-X. Jia, B.-H. Wang, Personal recommendation via unequal resource allocation on bipartite networks, Physica A 389 (16) (2010) 3282–3289.
[33] X.-K. Xu, S. Xu, Y.-X. Zhu, Q.-M. Zhang, Link predictability in complex networks, Complex Syst. Complex. Sci. 11 (2014) 41–47.
[34] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, H.E. Stanley, Toward link predictability of complex networks, Proc. Natl. Acad. Sci. 112 (8) (2015) 2325–2330.
[35] M.M. Dankulov, R. Melnik, B. Tadic, The dynamics of meaningful social interactions and the emergence of collective knowledge, Sci. Rep. 5 (2015) 12197 EP, Article.
[36] K.-I. Goh, G. Salvi, B. Kahng, D. Kim, Skeleton and fractal scaling in complex networks, Phys. Rev. Lett. 96 (2006) 018701.
[37] C. Song, Z. Qu, N. Blumm, A.-L. Barabási, Limits of predictability in human mobility, Science 327 (5968) (2010) 1018–1021.