This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2753818, IEEE Access



# Mixed Similarity Diffusion for Recommendation on Bipartite Networks

Ximeng Wang, Yun Liu, Guangquan Zhang, Yi Zhang, *Member, IEEE*, Hongshu Chen and Jie Lu, *Senior Member, IEEE*

***Abstract*—In recommender systems, collaborative filtering technology is an important method to evaluate user preference through exploiting user feedback data, and has been widely used in industrial areas. Diffusion-based recommendation algorithms inspired by diffusion phenomenon in physical dynamics are a crucial branch of collaborative filtering technology, which use a bipartite network to represent collection behaviors between users and items. However, diffusion-based recommendation algorithms calculate the similarity between users and make recommendations by only considering implicit feedback but neglecting the benefits from explicit feedback data, which would be a significant feature in recommender systems. This paper proposes a mixed similarity diffusion model to integrate both explicit feedback and implicit feedback. First, cosine similarity between users is calculated by explicit feedback and we integrate it with resource-allocation index calculated by implicit feedback. We further improve the performance of the mixed similarity diffusion model by considering the degrees of users and items at the same time in diffusion processes. Some sophisticated experiments are designed to evaluate our proposed method on three real-world datasets. Experimental results indicate that recommendations given by the mixed similarity diffusion perform better on both the accuracy and the diversity than that of most state-of-the-art algorithms.**

***Index Terms*—Recommender systems, diffusion processes, bipartite networks, collaborative filtering.**

## I. Introduction

WITH the revolutionary development of the Internet, the quantity of information is growing very quickly and has become out of the capability of human beings [1]. Information overload appears to be a serious problem in traditional data analytic studies [2]–[4], and exploring useful content from rapidly increasing information tends to be a raising trend in modern society [4]–[6]. Recommender systems have been recognized as an effective tool to handle this problem [7], and play a crucial role in data processing tasks. Recently, personalized recommendation among numerous potential choices attracts more and more attention [8], and have been applied to many actual domains, such as recommending movies [9], [10], content [11], citations [12], locations [13], [14], mobile applications [15] and services for e-business [16], [17] and e-goverment [18], [19].

Collaborative filtering is a typical and the most popular information filtering technology in recommender systems [20]. Its main idea is to evaluate user preference through exploiting user feedback data in a collective way. Two kinds of feedback data can be processed, i.e., explicit feedback and implicit feedback. The former, e.g., 5-star ratings, means the level of how a user likes an item, while the latter one, e.g., clicks or purchases, indicates whether a user likes an item or not [21]. In addition, elements in explicit feedback matrix can be any numeric values while the implicit feedback matrix is a single-valued matrix.

In collaborative filtering, diffusion-based recommendation algorithms are popular and make recommendations based on network structures, which are inspired by diffusion phenomenon in physical dynamics [22], [23]. These algorithms use a user-item bipartite network to represent input data, e.g., rating matrix, and links on the bipartite network indicate the collection behaviors between users and items. Some physical processes can be then employed on the bipartite network to make recommendations, such as random walk, mass diffusion [24] and heat conduction [25]. Unfortunately, since traditional diffusion-based recommendation methods use binary value to simulate user's collection behaviors, e.g., a user collects or rejects an item, those algorithms only take advantage of implicit feedback but neglect explicit feedback, which is also a crucial feature for precisely evaluating user preference [26], [27].

This paper proposes a two-step resource-allocation process to overcome the above research gap. A mixed similarity diffusion model is designed by involving both explicit feedback data and implicit feedback data, and inspired by the idea of hybrid diffusion [28], we consider the degrees of users and items at the same time in diffusion processes to improve the performance of the model. A series of experiments are conducted to evaluate the performance by comparing with several state-of-the-art diffusion-based recommendation algorithms on three real-world datasets. The results demonstrate that the mixed similarity diffusion model has better performance than most of baselines on both the accuracy and the diversity.

The remainder of this paper is organized as follows. Section 2 presents the related work of diffusion-based recommendation

X. Wang is with the Key Laboratory of Communication & Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, P.R. China and also with the Decision Systems & e-Service Intelligence Laboratory, Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia.

Y. Liu is with the Key Laboratory of Communication & Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, P.R. China.

G. Zhang, Y. Zhang and J. Lu are with the Decision Systems & e-Service Intelligence Laboratory, Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia.

H. Chen is with the Research and Innovation Office, University of Technology Sydney, NSW 2007, Australia.

Corresponding author: Y. Liu (E-mail: liuyun@bjtu.edu.cn)

Manuscript received August 9, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2753818, IEEE Access



algorithms and section 3 proposes the mixed similarity diffusion model. The description of three datasets and evaluation metrics are given in section 4. The experimental results and comparisons are presented in section 5, and section 6 concludes this paper with discussions and future work.

## II. Related Work

Collaborative filtering is one of the most widely used approaches in recommender systems and diffusion-based recommendation algorithms are a vital branch which based on bipartite networks or tripartite networks [29].

Diffusion-based recommendation algorithms are inspired by the diffusion phenomenon in physical dynamics, which simulates a resource-allocation process on bipartite user-item networks to make recommendations. In particular, some discussions about the bipartite user-item networks can be found in [30]. Mass diffusion (MD) [24] and heat conduction (HC) [25] are regarded as the pioneers of diffusion-based recommendation algorithms. These two approaches assume every item collected by a target user has one unit of initial resource, and build a two-step random walk process to redistribute the resource on bipartite networks [29]. Mass diffusion model distributes the resource based on each node's degree and focuses on the accuracy of recommendations, while heat conduction model reallocates the resource based on the neighboring nodes' degrees of each node to propose recommendations with high diversity.

Based on mass diffusion model, Zhou *et al.* improved both the accuracy and the diversity through eliminating redundant correlations between items in recommendation processes [31]; Chen *et al.* introduced cosine index between items into mass diffusion model to enhance the accuracy [32]; Lü *et al.* proposed a preferential diffusion model taking the heterogeneity of users' degrees into account [33]; Zeng *et al.* enhanced or suppressed the weight of similar users via a similarity-preferential diffusion process [34] and Wang *et al.* used a similarity bipartite network to make the resource-allocation process more personalized [35]. There are also some studies focusing on heat conduction model. Specifically, Liu *et al.* proposed a biased heat conduction model that decreases the attention of small-degree items to enhance both the accuracy and the diversity [36]. A combination model was proposed by Zhou *et al.*, which integrates ProbS with HeatS and seeks a way to balance the results with the accuracy and the diversity [28].

The configuration of initial resource distribution would significantly influence the performance of recommendation algorithms. Some previous studies suggest that decreasing the amount of initial resource on items with high degree can improve the accuracy of recommendations [37]. Moreover, additional features, e.g., tags and trust relations, can be utilized in diffusion processes through tripartite networks would also improve the performance and alleviate the cold-start problem in recommender systems [38], [39]. Time information plays an essential role in recommender systems, some time-aware approaches bring substantial improvements on the accuracy by reducing the impacts of out-of-date data [40], [41].

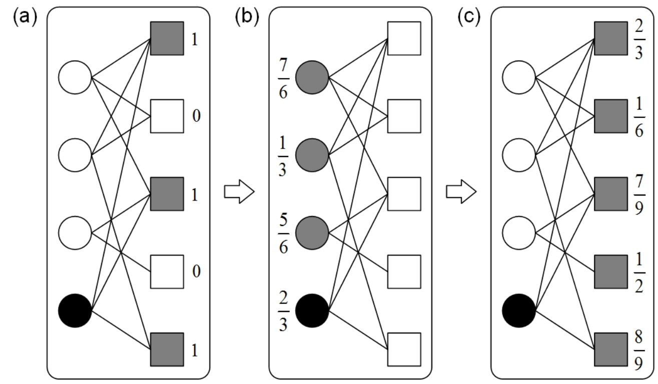


Fig. 1. An illustration of mass diffusion model (MD). Users and items are represented by circles and squares, respectively. The black circle means the target user, and the circles and squares with grey color indicate the resource is currently distributed on these nodes. Plot (a) is the initial configuration, each item linked to the target user obtains one unit of resource. Plot (b) shows that the resource flows from items to users according to each item's degree and the resource on each user can be calculated by Eq. (2). Plot (c) shows the resource flows back to items based on each user's degree and the final resource on each item is calculated by Eq. (3).

## III. Proposed Method

A recommender system can be represented by a user-item bipartite network which consists of a user set $U$ and an item set $I$. The user set is defined as $U = \{u_1, u_2, \dots, u_m\}$ and the item set is defined as $I = \{i_1, i_2, \dots, i_n\}$, where $m$ and $n$ are the numbers of users and items in the recommender system. A link set $E = \{e_1, e_2, \dots, e_z\}$ is used to denote relations between users and items and $z$ is the amount of links. In this paper, to make it easy to understand, we use Greek and Latin letters to express item-related and user-related indices, respectively. An $m \times n$ adjacent matrix $A$ can be utilized to describe the user-item bipartite network, where every element $a_{i\alpha}$ is defined in Eq. (1).

$$a_{i\alpha} = \begin{cases} 1, & \text{user } i \text{ collects item } \alpha, \\ 0, & \text{else.} \end{cases} \tag{1}$$

Degree is an important concept in complex networks, which is the number of edges linked to a vertex [24]. Accordingly, we define the degrees of item $\alpha$ and user $i$ as $k_\alpha$ and $k_i$ that represent the number of users who collect item $\alpha$ and the number of items collected by user $i$, respectively. The primary purpose of a ranking-based recommender system is to evaluate user preference and provide a recommendation list for a target user. That is to say, a set of items uncollected by the target user with the highest recommendation score would be included in the recommendation list. The length of the recommendation list is defined as $L$ in this paper.

### A. Mass Diffusion Model

Mass diffusion model (MD) is a successful and popular recommendation algorithm [24], which takes advantage of a resource-allocation process to make recommendations on a bipartite network. In MD, a target user who will receive a recommendation list of items needs to be chosen at first. Then, items linked to the target user on the bipartite network obtain initial resource. Note that we assume the initial resource on each item is one unit for convenient computation in this paper.

MD can be described as a two-step resource-allocation process. In step 1, the initial resource on item nodes flows to neighboring user nodes based on each item's degree, so the resource on each user node can be calculated as

$$f_{ij}^{'} = \sum_{\alpha=1}^{n} \frac{a_{i\alpha}a_{j\alpha}}{k_\alpha} f_\alpha, \tag{2}$$

where user $i$ is the target user and user $j$ is the user who will get resource in step 1; $f_\alpha$ and $f_{ij}^{'}$ are the initial resource on item $\alpha$ and the resource on user $j$ after step 1, respectively; $a_{i\alpha}$ and $a_{j\alpha}$ are elements in the adjacent matrix $A$, and $k_\alpha$ is the degree of item $\alpha$. The distribution strategy in step 2 is based on each user's degree. Therefore, the final resource $f_{i\beta}^{'}$ on item $\beta$ is defined as

$$f_{i\beta}^{'} = \sum_{j=1}^{m} \frac{a_{j\beta}}{k_j} f_{ij}^{'}, \tag{3}$$

where $k_j$ is the degree of user $j$. After a two-step resource-allocation process, the initial resource is redistributed on items, and then a recommendation list of uncollected items can be arranged for the target user according to the final resource on each item. The uncollected items with most final resource will be placed at the top of the list. An illustration of the resource-allocation process of the mass diffusion model is shown in Figure 1.

Although MD is proved to be effective in recommendation tasks, there are still some weaknesses. Firstly, MD makes recommendations with implicit feedback which only includes binary value, such as 1 for a positive example and 0 for a negative example. Explicit feedback is neglected by MD. However, explicit feedback is also very important in recommender systems. For example, ratings are multivariant in real-world datasets, e.g., a user can give a rating to an item from 1 to 5, which can be regarded as explicit feedback. Furthermore, MD only considers the degree of each user when the resource flows back to items in step 2. In some previous studies [25], [36], the degree of each item plays a vital role in the resource redistribution process, which can improve the diversity of recommendations.

In this paper, we will improve mass diffusion model through a mixed similarity diffusion strategy that integrates both implicit feedback and explicit feedback. Additionally, we will consider the degrees of users and items at the same time when the resource flows on the bipartite network.

### B. Similarity Measurement Methods

The similarity measurement between users is a crucial part in evaluating user preference. In recommender systems, we always assume a user will accept suggestions or choices from other most similar users. Therefore, how to measure the similarity between users obtains a lot of attention recently. In our method, we take advantage of two common similarity measurement methods, i.e., cosine similarity and resource-allocation (RA) index [32], to integrate implicit feedback and explicit feedback into the diffusion process.

The cosine similarity is a widely used approach in evaluating user preference based on explicit feedback, e.g., ratings. Between user $i$ and $j$, the cosine similarity is defined as

$$Cos(i,j) = \frac{\sum_{\alpha=1}^{n^{'}} R_{i\alpha}R_{j\alpha}}{\sqrt{\sum_{\alpha=1}^{n^{'}} R_{i\alpha}^2}\sqrt{\sum_{\alpha=1}^{n^{'}} R_{j\alpha}^2}}, \tag{4}$$

where $R_{i\alpha}$ and $R_{j\alpha}$ are rating scores on item $\alpha$ rated by user $i$ and $j$; $n^{'}$ is the number of co-rated items by both users. The value of cosine similarity is located in $[0,1]$, because the rating scores are greater than 0. The cosine similarity measures the angle between two user vectors of ratings, where a greater value of the cosine similarity indicates the closer relationship between two users [42].

The RA index is a typical similarity measurement on bipartite networks. The usual configuration of initial resource on each node is binary, i.e., 0 and 1, which resembles implicit examinations such as clicks, browses and collections in real systems. Thus, evaluating the similarity between two nodes via the RA index is to calculate the similarity with implicit feedback in recommender systems. The similarity between two nodes $i$ and $j$ can be defined as

$$RA(i,j) = \sum_{\alpha=1}^{n} \frac{a_{i\alpha}a_{j\alpha}}{k_\alpha}, \tag{5}$$

where $a_{i\alpha}$ and $a_{j\alpha}$ are the elements in adjacent matrix $A$; $k_\alpha$ is the degree of node $\alpha$.

If we assume the nodes $i$ and $j$ represent two users and the node $\alpha$ represents an item, it becomes a part of step 1 in MD model. The resource-allocation process then can be regarded as a one-step random walk on the user-item bipartite network starting from their common neighbors. So step 1 in MD model is equivalent to a similarity measurement process between two users when the initial resource on items is one unit.

### C. Mixed Similarity Diffusion for Recommendation

A mixed similarity diffusion model (MSD) is proposed by integrating both explicit feedback and implicit feedback. In MD, the resource is distributed based on each node's degree, which leads to non-personalized recommendations. While step 1 in MD only considers implicit feedback, MSD involves explicit feedback together. Similarly, a two-step resource-allocation process for MSD model is given below.

Step 1: We assume each item collected by the target user $i$ is assigned with one unit of initial resource. So, the amount of resource will be distributed to user $j$ is defined as

$$f_{ij}^{'} = \sum_{\alpha=1}^{n} \frac{a_{i\alpha}a_{j\alpha}Cos(i,j)}{\sum_{k=1}^{m} a_{k\alpha}Cos(i,k)} f_\alpha, \tag{6}$$

where $Cos(i,j)$ and $Cos(i,k)$ are the cosine similarity calculated by explicit feedback, e.g., ratings; $\sum_{k=1}^{m} a_{k\alpha}Cos(i,k)$ means the sum of similarity between the target user $i$ and all users who have collected the item $\alpha$, which is a normalization. In this step, we integrate the cosine similarity and RA index to propose a resource-allocation strategy based on mixed similarity and two kinds of feedback are both used for calculating the mixed similarity between users in the resource-allocation process.

Step 2: The resource allocated on users will flow back to items, in order to finish the resource redistribution process. We intend to consider both user's degree and item's degree to enhance the diversity of recommendations. A parameter $\lambda$ is introduced into our model to control the impact of user's degree and item's degree in this step. Assuming item $\beta$ will receive the resource from users, the final resource on item $\beta$ can be defined as

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{k_\beta^\lambda k_j^{1-\lambda}} f'_{ij}, \tag{7}$$

where we substitute Eq. (6) into Eq. (7) to generate the final model of our proposed method and project the resource-allocation process onto an item-item network, as

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{k_\beta^\lambda k_j^{1-\lambda}} \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha} Cos(i,j)}{\sum_{k=1}^{m} a_{k\alpha} Cos(i,k)} f_\alpha. \tag{8}$$

Finally, all items are sorted by their final resource and then a top-$L$ recommendation list of uncollected items is generated for the target user $i$. The pseudo-code of mixed similarity diffusion method is shown Algorithm 1.

**Algorithm 1** The algorithm of mixed similarity diffusion (MSD).

**Input:** An adjacent matrix $\boldsymbol{A}^{m\times n}$, a cosine similarity matrix $\boldsymbol{Cos}^{m\times m}$ and a parameter $\lambda$.

1: Initialization of a final resource vector $\boldsymbol{V}^{1\times n}$ for the target user $i$.
2: **for** $\beta = 1, 2, \ldots, n$ **do**
3: Set $f'_{i\beta} = 0$.
4: **for** $j = 1, 2, \ldots, m$ **do**
5: Set $f'_{ij} = 0$.
6: **for** $\alpha = 1, 2, \ldots, n$ **do**
7: Calculate $f'_{ij} = f'_{ij} + \frac{a_{i\alpha} a_{j\alpha} Cos(i,j)}{\sum_{k=1}^{m} a_{k\alpha} Cos(i,k)} f_\alpha$.
8: **end for**
9: Calculate $f'_{i\beta} = f'_{i\beta} + \frac{a_{j\beta}}{k_\beta^\lambda k_j^{1-\lambda}} f'_{ij}$.
10: **end for**
11: Set $V(\beta) = f'_{i\beta}$.
12: **end for**

**Output:** A recommendation list for the target user $i$, which is generated by descending order of the final resource on uncollected items in $\boldsymbol{V}$.

## IV. Data and Evaluation Metrics

This section describes the details of three benchmark datasets at first, and then a series of evaluation metrics are presented.

### A. Data Description

We use three different versions of MovieLens[1] datasets in our experiments including ML100K, ML1M and MLlatest to evaluate our proposed method in different circumstances. MovieLens datasets are public and real-world datasets, which are widely used for evaluating the performance of algorithms in recommender systems. The ML100K consists of 943 users, 1682 items and 100,000 observed ratings, while the ML1M has 1,000,209 ratings of 6040 users and 3706 items. In addition, we extract 94,537 ratings of 4801 items which are at least collected by three users from the MovieLens latest dataset (MLlatest) published on September, 2016. The range of ratings in MovieLens datasets is $[1,5]$. Statistics of these three datasets are illustrated in Table 1.

[1]https://grouplens.org/datasets/movielens/

TABLE I
Statistics of Datasets

| Dataset | Users | Items | Ratings | Sparsity |
|---|---|---|---|---|
| ML100K | 943 | 1682 | 100,000 | $6.30\times10^{-2}$ |
| ML1M | 6040 | 3706 | 1,000,209 | $4.47\times10^{-2}$ |
| MLlatest | 671 | 4801 | 94,537 | $2.93\times10^{-2}$ |

Following some previous studies [24], [35], we convert ratings to binary links to build a bipartite network where we assign 1 as 'relevant' for the ratings above 3 and 0 as 'non-relevant' for the remaining ratings. Note that the cosine similarity between users is calculated by the original ratings of all datasets, because it evaluates user's preference by explicit feedback.

A five-fold cross-validation is utilized in our experiments. We randomly divide each dataset into five folds and four are regarded as the training set, with the remaining fold treated as the testing set. Five iterations are arranged to make sure that all folds are tested.

### B. Evaluation Metrics

To present a comprehensive evaluation of the recommendation performance, we take advantage of some widely used evaluation metrics to measure the accuracy and diversity of our proposed method. The following metrics are used to measure the accuracy of recommendations.

**Precision** (*Pre@L*) is an important evaluation metric for ranking prediction in recommender systems, which measures the fraction of top-$L$ recommended items that are consumed by the target user. Mathematically, the average value of *Pre@L* for all users is defined as

$$Pre@L = \frac{1}{m}\left(\sum_{i=1}^{m} \frac{D_i(L)}{L}\right), \tag{9}$$

where $D_i(L)$ is the number of recommended items consumed by user $i$ in test set when the length of the recommendation list is $L$.

**Recall** (*Rec@L*) is another crucial metric in recommender systems, which calculates the proportion of correct recommended items and the number of total items in the test set for the target user. The average value of *Rec@L* for all users is defined as

$$Rec@L = \frac{1}{m}\left(\sum_{i=1}^{m} \frac{D_i(L)}{T_i(L)}\right), \tag{10}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2753818, IEEE Access



where $T_i(L)$ is the number of items collected in the test set.

**Rank-Biased Precision** (*RBP@L*) [43] assumes each user has a fixed probability $p$ to scan next recommended item from the first place in a recommendation list, defined as

$$RBP@L = \frac{1}{m}\left(\sum_{i=1}^{m}(1-p)\sum_{\alpha=1}^{L} c_{i\alpha}p^{\alpha-1}\right), \tag{11}$$

where $c_{i\alpha} = 1$ means that the $\alpha$th item in the recommendation list $L$ is collected by user $i$ in test set and $c_{i\alpha} = 0$ is the opposite. *RBP@L* is a significant ranking-based measurement that is very needful, because users always accept the recommended items at the top of a recommendation list and *RBP@L* is very close to an individual's actual habits of collecting items. In this paper, we assume the probability $p$ is 0.5.

**Mean Reciprocal Rank** (*MRR*) directly utilizes the reciprocal of the items position in a recommendation list to measure the performance of recommendation algorithms, defined as

$$MRR = \frac{1}{m}\left(\sum_{i=1}^{m}\sum_{\alpha\in S(i)}\frac{1}{rank_{\alpha}^{i}}\right), \tag{12}$$

where $S(i)$ is the items collected by user $i$ in test set and $rank_{\alpha}^{i}$ is the position of item $i$ in the recommendation list for user $i$. Be similar to *RBP@L*, *MRR* is also a ranking-based measurement that supposes the item consumed by the target user in test set placed at the top of the recommendation list obtains a grater score in the evaluation than the item at the bottom. Therefore, a larger value of *MRR* means a better performance.

The diversity also plays an important role in recommender systems, which indicates the ability of pushing out unpopular items for users. A metric used to measure the diversity of recommendations is represented as follows.

**Hamming Distance** (*Ham@L*) is a common method to evaluate the diversity of recommendations. The definition of *Ham@L* for all users is

$$Ham@L = \frac{1}{m(m-1)}\sum_{i\neq j}\left(1-\frac{Q_{ij}(L)}{L}\right), \tag{13}$$

where $Q_{ij}(L)$ is the number of overlapped items in the recommendation lists for user $i$ and user $j$. The larger value of *Ham@L* means the higher diversity.

## V. EXPERIMENTS

This section introduces the baselines that will be used for comparing with our method, the impact of parameter $\lambda$ , and the results of the comparative experiments.

### A. Baselines

We intend to compare the performance of our proposed method with some classic baselines to verify the superiority.

**PopRank** is a basic recommendation algorithm with implicit feedback, which provides a recommendation list based on items' popularity. The most popular item will be arranged at the top. In our experiments, we regard the degree of each item as its popularity. The item with larger degree means it has already collected by more users, i.e., grater popularity.

**UserCF** is a classic collaborative filtering method based on the cosine similarity between users. This method assumes the target user will accept the opinions from the most similar users.

**MD** [24] is a pioneer of diffusion-based recommendation algorithms, which uses a resource-allocation process to make recommendations on bipartite networks.

**HC** [25] employs the heat conduction process of physical dynamics on recommendation tasks. This model is good at pushing out small-degree items, so the results have a high diversity in general.

**CosRA** [32] is a vertex similarity index on bipartite networks. The recommendation algorithm based on CosRA index is named CosRA-based method that can be regarded as a special situation of hybrid diffusion [28].

**SPMD** [34] is an improvement on mass diffusion model. It introduces similarity-preferential diffusion into the recommendation process, which can enhance or suppress the weight of users who are most similar to the target user.

**BHC** [36] is an improvement on heat conduction model. Because heat conduction model sacrifices the accuracy of recommendations to push out small-degree items, BHC proposes a biased resource distribution strategy to enhance the precision of recommendations.

### B. The Impact of Parameter $\lambda$

In the mixed similarity diffusion model, the parameter $\lambda$ controls the impact of user's degree and item's degree in the second step of the resource-allocation process. To determine the optimal value of $\lambda$ in our method, we adjust the parameter on all three datasets. The precision and recall can comprehensively reflect the accuracy, so we use these two metrics to determine the optimal value of $\lambda$ for MSD when the recommendation list is $L = 10$.

Figure 2 reports the results of *Pre@10* and *Rec@10* for our proposed method when the parameter $\lambda$ changes from 0 to 1 at a calculative step of 0.05. Figure 2(a), (b) and (c) represent the variation of *Pre@10* and *Rec@10* in ML100K, ML1M and MLlatest datasets, respectively. It can be seen from Figure 2, the optimal value of the parameter $\lambda$ is 0.55, 0.6 and 0.5 for ML100K, ML1M and MLlatest, respectively. Even though the sparsity of these three datasets is totally different, the observation that the optimal $\lambda$ locates around 0.55 might support the inference that our proposed model has certain practical value. Generally, if $\lambda = 0$, our model becomes a simple one that only combines MD model with the cosine similarity with explicit feedback. According to the experiment results, with the increasing value of parameter $\lambda$, the degree of items provides more impact on the final recommendations and improve the accuracy.

### C. Recommendation Performance Evaluation

In this paper, we use three real-world rating datasets to evaluate our proposed method that is compared with seven baselines. The whole experiment results are presented in Table 2 and the optimal parameters of every algorithm in different datasets are also included in this table for result reproducibility.

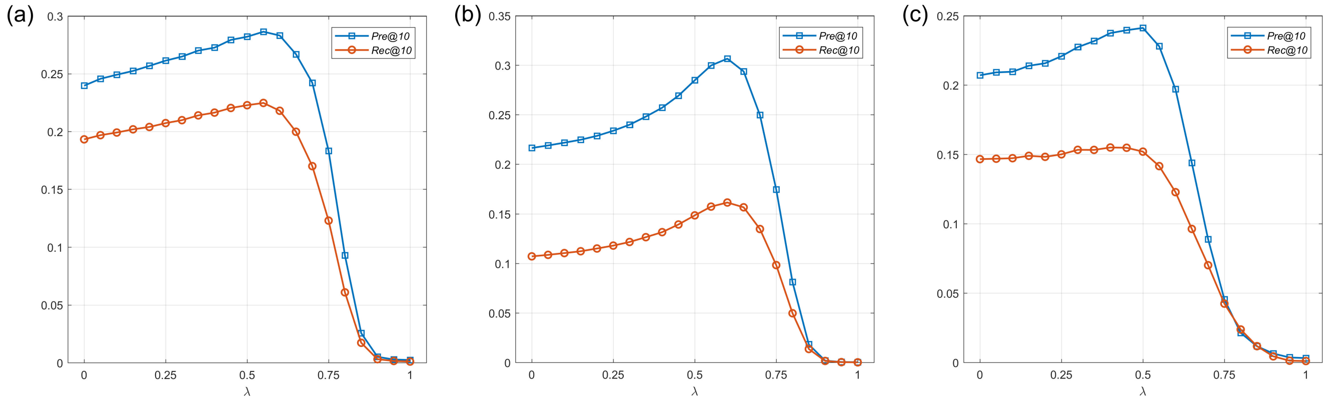


Fig. 2. The *Pre@10* and *Rec@10* of MSD when changing the parameter $\lambda$ between 0 and 1 at a calculative step of 0.05 in ML100K, ML1M and MLlatest datasets are represented. (a) The optimal value is $\lambda = 0.55$ in ML100K. (b) The optimal value is $\lambda = 0.6$ in ML1M. (c) The optimal value is $\lambda = 0.5$ in MLlatest.

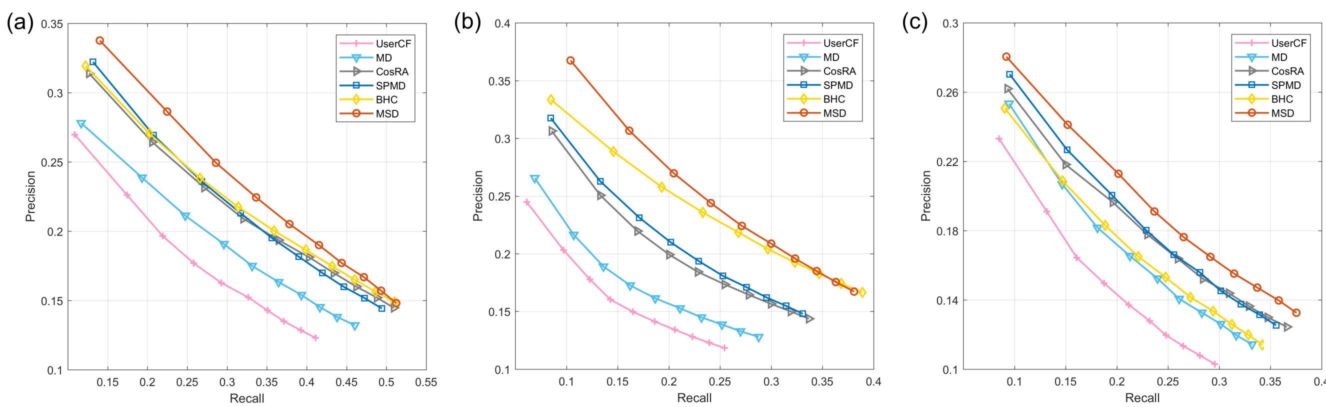


Fig. 3. The Precision-Recall curves on ML100K, ML1M and MLlatest datasets are represented in diagrams (a), (b) and (c), respectively, where the length of recommendations is from 5 to 50 at a calculative step of 5. Because of the poor performance of PopRank and HC in these two metrics, we do not show their results in this figure.

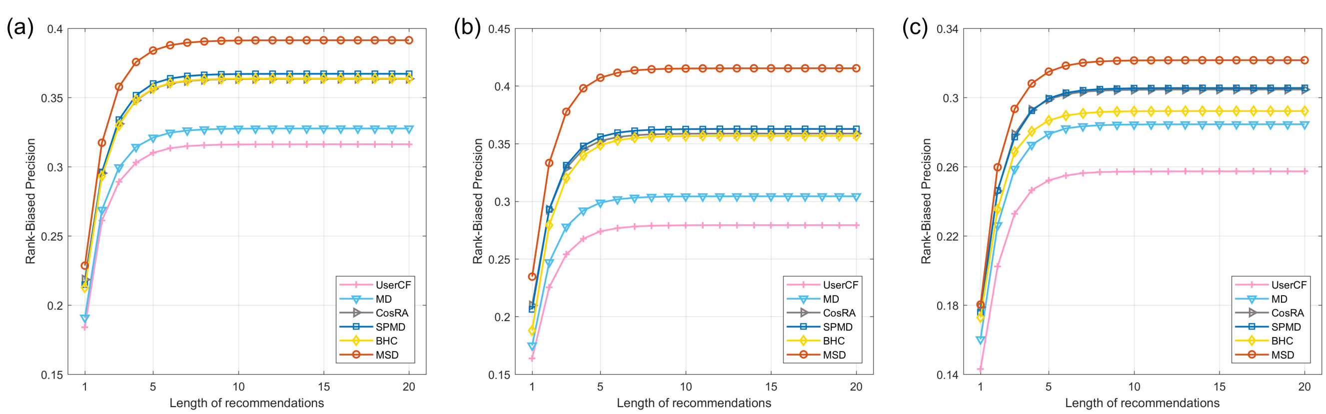


Fig. 4. The Rank-Biased Precision ($p = 0.5$) of MSD and five baselines on ML100K, ML1M and MLlatest datasets are represented in diagrams (a), (b) and (c), respectively, where the length of recommendations is from 1 to 20. Because of the poor performance of PopRank and HC in this metric, we do not show their results in this figure.

TABLE II
RECOMMENDATION PERFORMANCE OF MSD AND SEVEN BASELINES ON ML100K, ML1M AND MLLATEST DATASETS

| ML100K | *Pre@5* | *Pre@10* | *Rec@5* | *Rec@10* | *RBP@5* | *RBP@10* | *MRR* | *Ham@5* | *Ham@10* |
|---|---|---|---|---|---|---|---|---|---|
| PopRank | 0.1639 | 0.1560 | 0.0567 | 0.1106 | 0.2014 | 0.2061 | 0.7664 | 0.5126 | 0.4601 |
| UserCF | 0.2698 | 0.2261 | 0.1086 | 0.1743 | 0.3101 | 0.3161 | 1.0607 | 0.7319 | 0.6769 |
| MD | 0.2781 | 0.2387 | 0.1167 | 0.1932 | 0.3210 | 0.3275 | 1.1114 | 0.7534 | 0.7150 |
| HC | 0.0025 | 0.0036 | 0.0013 | 0.0032 | 0.0015 | 0.0016 | 0.1702 | 0.8852 | 0.8686 |
| CosRA | 0.3139 | 0.2644 | 0.1267 | 0.2059 | 0.3563 | 0.3634 | 1.2229 | 0.8378 | 0.8082 |
| SPMD ($\theta = 2.25$) | 0.3222 | 0.2693 | 0.1316 | 0.2077 | 0.3601 | 0.3670 | 1.2317 | 0.8608 | 0.8434 |
| BHC ($\lambda = 0.75$) | 0.3194 | 0.2704 | 0.1223 | 0.2058 | 0.3565 | 0.3636 | 1.2380 | **0.9109** | 0.8774 |
| MSD ($\lambda = 0.55$) | **0.3376** | **0.2863** | **0.1401** | **0.2247** | **0.3840** | **0.3912** | **1.2956** | **0.9109** | **0.8849** |
| ML1M | *Pre@5* | *Pre@10* | *Rec@5* | *Rec@10* | *RBP@5* | *RBP@10* | *MRR* | *Ham@5* | *Ham@10* |
| PopRank | 0.1939 | 0.1681 | 0.0444 | 0.0736 | 0.2091 | 0.2135 | 0.8160 | 0.5128 | 0.4729 |
| UserCF | 0.2448 | 0.2032 | 0.0612 | 0.0970 | 0.2740 | 0.2792 | 0.9927 | 0.6326 | 0.5670 |
| MD | 0.2656 | 0.2164 | 0.0690 | 0.1070 | 0.2988 | 0.3042 | 1.0688 | 0.6926 | 0.6103 |
| HC | 0.0007 | 0.0026 | 0.0003 | 0.0020 | 0.0004 | 0.0005 | 0.2248 | 0.8812 | 0.8470 |
| CosRA | 0.3063 | 0.2507 | 0.0856 | 0.1330 | 0.3524 | 0.3586 | 1.2262 | 0.7965 | 0.7197 |
| SPMD ($\theta = 3.05$) | 0.3175 | 0.2626 | 0.0847 | 0.1331 | 0.3560 | 0.3626 | 1.2547 | 0.8670 | 0.8166 |
| BHC ($\lambda = 0.85$) | 0.3334 | 0.2885 | 0.0849 | 0.1458 | 0.3486 | 0.3566 | 1.3089 | 0.9315 | 0.9011 |
| MSD ($\lambda = 0.6$) | **0.3674** | **0.3066** | **0.1040** | **0.1614** | **0.4072** | **0.4152** | **1.4234** | **0.9440** | **0.9126** |
| MLlatest | *Pre@5* | *Pre@10* | *Rec@5* | *Rec@10* | *RBP@5* | *RBP@10* | *MRR* | *Ham@5* | *Ham@10* |
| PopRank | 0.1699 | 0.1364 | 0.0523 | 0.0771 | 0.1815 | 0.1849 | 0.6676 | 0.4862 | 0.4323 |
| UserCF | 0.2331 | 0.1911 | 0.0850 | 0.1315 | 0.2521 | 0.2572 | 0.8883 | 0.6373 | 0.6184 |
| MD | 0.2534 | 0.2069 | 0.0947 | 0.1465 | 0.2789 | 0.2844 | 0.9768 | 0.7153 | 0.6868 |
| HC | 0.0155 | 0.0168 | 0.0059 | 0.0125 | 0.0114 | 0.0119 | 0.1524 | 0.9332 | **0.9227** |
| CosRA | 0.2620 | 0.2182 | 0.0931 | 0.1499 | 0.2989 | 0.3045 | 1.0440 | 0.8515 | 0.8132 |
| SPMD ($\theta = 2.1$) | 0.2703 | 0.2268 | **0.0952** | 0.1514 | 0.2995 | 0.3054 | 1.0550 | 0.8622 | 0.8243 |
| BHC ($\lambda = 0.5$) | 0.2507 | 0.2088 | 0.0906 | 0.1468 | 0.2868 | 0.2921 | 0.9905 | 0.8197 | 0.7757 |
| MSD ($\lambda = 0.5$) | **0.2805** | **0.2411** | 0.0921 | **0.1520** | **0.3150** | **0.3214** | **1.1083** | **0.9413** | 0.9157 |

In ML100K, MSD obtains the best results in all metrics of accuracy, i.e., Precision, Recall and Rank-Biased Precision, when the length of the recommendation list is 5 and 10. Compared to SPMD, *Pre@5*, *Rec@5*, *RBP@5* and *MRR* can be improved 4.8%, 6.5%, 6.6% and 5.2% by MSD, respectively. The results of MSD and BHC are same on *Ham@5*, however, MSD gets a better result on *Ham@10*. In ML1M, all the best results on the accuracy and the diversity are achieved by MSD that enhances *Pre@5*, *Rec@5*, *RBP@5* and *MRR* by 10.2%, 22.5%, 16.8% and 8.7% than BHC. MSD also brings an 1.3% improvement on *Ham@5* than BHC which gets the best on diversity in seven baselines. Furthermore, MSD achieves the best results on most of metrics in MLlatest dataset.

Figure 3 depicts the Precision-Recall curves on three datasets with the length of the recommendation list from 5 to 50 at a calculative step of 5. In Figure 3(a) and (c), MSD always has the best results. In Figure 3(b), MSD gets the best performance when the recommendation list is short. As the length increases, BHC gradually obtains better performance than MSD. However, a user always pay more attention to the items at the top of a recommendation list, so top-10 recommendations are the most important in evaluating recommender systems, which means the algorithm with the best performance in the short recommendation list is more meaningful for practical applications. The results of Rank-Biased Precision are shown in Figure 4, MSD always keeps the best results in three benchmark datasets.

The accuracy-diversity dilemma is ubiquitous in recommender systems [28]. A popular item should be accepted by most users, so recommending a list of popular items to a user in accordance with his/her preference may enhance the accuracy but reduce the diversity. Figure 5 indicates the results of Hamming Distance. HC focuses on pushing out small-degree items which lead to a high diversity with the increasing of the recommendation list. However, considering the poor performance of HC on the accuracy, it is hard to apply HC in real-world systems. When the recommendation list is short, MSD surpasses HC and proposes recommendations with higher diversity. Therefore, our proposed method improves the accuracy and the diversity at the same time, which means it relieves this dilemma to some extent and makes an accuracy-diversity balance in recommender systems.

## VI. CONCLUSION AND FUTURE WORK

This paper proposes a mixed similarity diffusion model (MSD) to improve the performance of recommendation, which integrates the similarity from both explicit feedback and implicit feedback. We calculate the cosine similarity with explicit

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2753818, IEEE Access



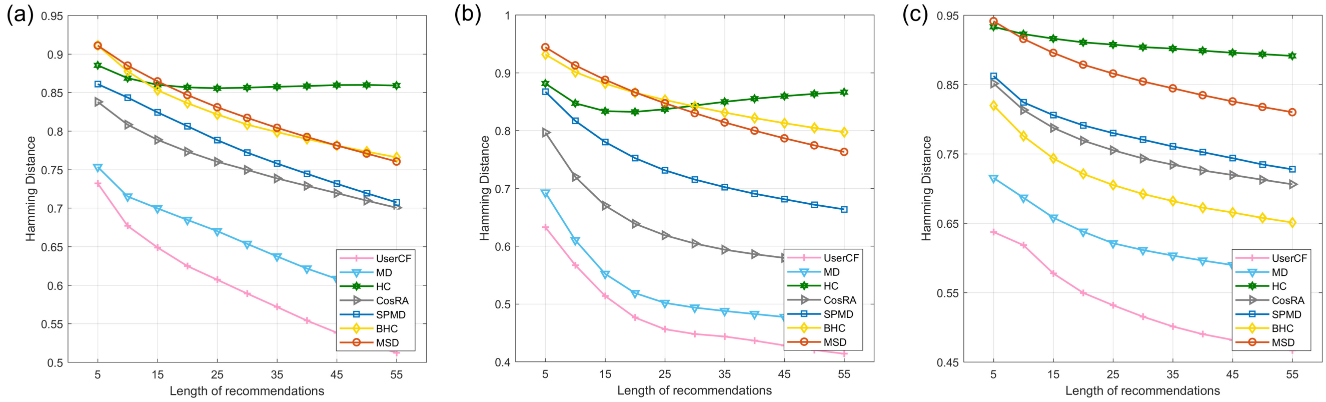


Fig. 5. The Hamming Distance of MSD and six baselines on ML100K, ML1M and MLlatest datasets are represented in diagrams (a), (b) and (c), respectively, where the length of recommendations is from 5 to 55 at a calculative step of 5. Because of the poor performance of PopRank in this metric, we do not show its results in this figure.

feedback data and the resource-allocation index with implicit feedback data, and combine these two kinds of similarity into diffusion processes. Experiments on three real-world datasets demonstrate our method performs better than most baselines. Specifically, MSD brings some significant improvements compared to BHC, SPMD and CosRA which are the state-of-the-art diffusion-based recommendation algorithms. MSD also proposes recommendations with higher diversity than HC when the recommendation list is short. Therefore, MSD achieves an accuracy-diversity balance, which enhances the accuracy and the diversity at the same time.

For future work, we are interested in studying the diffusion-based recommendation on some extremely sparse datasets. Moreover, we will continue our research on improving the performance of the diffusion-based recommendation via some additional features, e.g., time information [40], [41] and social trust [44], [45]. We hope this paper will inspire readers in this significant direction.

## ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61401015, the Fundamental Research Funds for the Central Universities under Grant Nos. 2017JBM013 and W17JB00060, and the Australian Research Council under Grant No. DP170101632.

## REFERENCES

[1] R. T. Ma, J. Lui, and V. Misra, "Evolution of the internet economic ecosystem," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 85–98, 2015.

[2] Y. Zhang, G. Zhang, D. Zhu, and J. Lu, "Scientific evolutionary pathways: Identifying and visualizing relationships for scientific topics," *Journal of the Association for Information Science and Technology*, vol. 68, no. 8, pp. 1925–1939, 2017.

[3] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 744–758, 2017.

[4] H. Wang, P. Zhang, X. Zhu, I. W.-H. Tsang, L. Chen, C. Zhang, and X. Wu, "Incremental subgraph feature selection for graph classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 128–142, 2017.

[5] Y. Zhang, G. Zhang, H. Chen, A. L. Porter, D. Zhu, and J. Lu, "Topic analysis and forecasting for science, technology and innovation: Methodology with a case study focusing on big data research," *Technological Forecasting and Social Change*, vol. 105, pp. 179–191, 2016.

[6] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 1895–1901.

[7] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[8] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.

[9] A. Gogna and A. Majumdar, "A comprehensive recommender system model: Improving accuracy for both warm and cold start users," *IEEE Access*, vol. 3, pp. 2803–2813, 2015.

[10] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decision Support Systems*, vol. 87, pp. 80–93, 2016.

[11] C. Verma, M. Hart, S. Bhatkar, A. Parker-Wood, and S. Dey, "Improving scalability of personalized recommendation systems for enterprise knowledge workers," *IEEE Access*, vol. 4, pp. 204–215, 2016.

[12] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, "Context-based collaborative filtering for citation recommendation," *IEEE Access*, vol. 3, pp. 1695–1703, 2015.

[13] H. Wang, Y. Fu, Q. Wang, H. Yin, C. Du, and H. Xiong, "A location-sentiment-aware recommender system for both home-town and out-of-town users," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1135–1143.

[14] L. Wang, Z. Yu, B. Guo, T. Ku, and F. Yi, "Moving destination prediction using sparse dataset: A mobility gradient descent approach," *ACM Transactions on Knowledge Discovery from Data*, vol. 11, no. 3, p. 37, 2017.

[15] H. Yin, L. Chen, W. Wang, X. Du, Q. V. H. Nguyen, and X. Zhou, "Mobi-sage: A sparse additive generative model for mobile app recommendation," in *Proceedings of the 33rd International Conference on Data Engineering*, 2017, pp. 75–78.

[16] D. Wu, G. Zhang, and J. Lu, "A fuzzy preference tree-based recommender system for personalized business-to-business e-services," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 29–43, 2015.

[17] J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, "a web-based personalized business partner recommendation system using fuzzy semantic techniques," *Computational Intelligence*, vol. 29, no. 1, pp. 37–69, 2013.

[18] D. Wu, J. Lu, and G. Zhang, "A fuzzy tree matching-based personalized e-learning recommender system," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2412–2426, 2015.

[19] J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, "Bizseeker: a hybrid semantic recommendation system for personalized government-to-business e-services," *Internet Research*, vol. 20, no. 3, pp. 342–365, 2010.

[20] Z. Xu, F. Zhang, W. Wang, H. Liu, and X. Kong, "Exploiting trust and

usage context for cross-domain recommendation," *IEEE Access*, vol. 4, pp. 2398–2407, 2016.

[21] M. Liu, W. Pan, M. Liu, Y. Chen, X. Peng, and Z. Ming, "Mixed similarity learning for recommendation with implicit feedback," *Knowledge-Based Systems*, vol. 119, pp. 178–185, 2017.

[22] F. Xiong and Z.-Y. Li, "Effective methods of restraining diffusion in terms of epidemic dynamics," *Scientific Reports*, vol. 7, p. 6013, 2017.

[23] H. Wang, J. Wu, S. Pan, P. Zhang, and L. Chen, "Towards large-scale social networks with online diffusion provenance detection," *Computer Networks*, vol. 114, pp. 154–166, 2017.

[24] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, p. 046115, 2007.

[25] Y.-C. Zhang, M. Blattner, and Y.-K. Yu, "Heat conduction process on community networks as a recommendation model," *Physical Review Letters*, vol. 99, no. 15, p. 154301, 2007.

[26] W. Pan and Z. Ming, "Collaborative recommendation with multiclass preference context," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 45–51, 2017.

[27] H. Wang, J. Wu, C. Zhou, Z. Ji, and J. Wu, "Mining subcascade features for cascade outbreak prediction in big networks," in *Proceedings of the 29th International Joint Conference on Neural Networks*, 2016, pp. 3942–3949.

[28] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.

[29] F. Yu, A. Zeng, S. Gillard, and M. Medo, "Network-based recommendation algorithms: A review," *Physica A: Statistical Mechanics and its Applications*, vol. 452, pp. 192–208, 2016.

[30] M.-S. Shang, L. Lü, Y.-C. Zhang, and T. Zhou, "Empirical analysis of web-based user-object bipartite networks," *Europhysics Letters*, vol. 90, no. 4, p. 48006, 2010.

[31] T. Zhou, R.-Q. Su, R.-R. Liu, L.-L. Jiang, B.-H. Wang, and Y.-C. Zhang, "Accurate and diverse recommendations via eliminating redundant correlations," *New Journal of Physics*, vol. 11, no. 12, p. 123008, 2009.

[32] L.-J. Chen, Z.-K. Zhang, J.-H. Liu, J. Gao, and T. Zhou, "A vertex similarity index for better personalized recommendation," *Physica A: Statistical Mechanics and its Applications*, vol. 466, pp. 607–615, 2017.

[33] L. Lü and W. Liu, "Information filtering via preferential diffusion," *Physical Review E*, vol. 83, no. 6, p. 066119, 2011.

[34] A. Zeng, A. Vidmer, M. Medo, and Y.-C. Zhang, "Information filtering by similarity-preferential diffusion processes," *Europhysics Letters*, vol. 105, no. 5, p. 58002, 2014.

[35] X. Wang, Y. Liu, and F. Xiong, "Improved personalized recommendation based on a similarity network," *Physica A: Statistical Mechanics and its Applications*, vol. 456, pp. 271–280, 2016.

[36] J.-G. Liu, T. Zhou, and Q. Guo, "Information filtering via biased heat conduction," *Physical Review E*, vol. 84, no. 3, p. 037101, 2011.

[37] T. Zhou, L.-L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *Europhysics Letters*, vol. 81, no. 5, p. 58004, 2008.

[38] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 179–186, 2010.

[39] M.-S. Shang, Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Collaborative filtering with diffusion-based similarity on tripartite graphs," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 6, pp. 1259–1264, 2010.

[40] A. Vidmer and M. Medo, "The essential role of time in network-based recommendation," *Europhysics Letters*, vol. 116, no. 3, p. 30007, 2016.

[41] F. Zhang, Q. Liu, and A. Zeng, "Timeliness in recommender systems," *Expert Systems with Applications*, vol. 85, pp. 270–278, 2017.

[42] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[43] A. Moffat and J. Zobel, "Rank-biased precision for measurement of retrieval effectiveness," *ACM Transactions on Information Systems*, vol. 27, no. 1, pp. 2–27, 2008.

[44] F. Xiong, Y. Liu, and J. Cheng, "Modeling and predicting opinion formation with trust propagation in online social networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 44, pp. 513–524, 2017.

[45] X. Wang, Y. Liu, G. Zhang, F. Xiong, and J. Lu, "Diffusion-based recommendation with trust relations on tripartite graphs," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2017, no. 8, p. 083405, 2017.