

A multilayer network diffusion-based model for reviewer recommendation

Yiwei Huang(黄羿炜), Shuqi Xu(徐舒琪), Shimin Cai(蔡世民), and Linyuan Lü(吕琳媛)
Citation: Chin. Phys. B, 2024, 33 (**3**): 038901. DOI: 10.1088/1674-1056/ad181d
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Source localization in signed networks with effective distance

Zhi-Wei Ma(马志伟), Lei Sun(孙蕾), Zhi-Guo Ding(丁智国), Yi-Zhen Huang(黄宜真), and Zhao-Long Hu(胡兆龙)
Chin. Phys. B, 2024, 33 (**2**): 028902. DOI: 10.1088/1674-1056/ad1482

Identifying multiple influential spreaders in complex networks based on spectral graph theory

Dong-Xu Cui(崔东旭), Jia-Lin He(何嘉林), Zi-Fei Xiao(肖子飞), and Wei-Ping Ren(任卫平)
Chin. Phys. B, 2023, 32 (**9**): 098904. DOI: 10.1088/1674-1056/acac0b

Self-similarity of complex networks under centrality-based node removal strategy

Dan Chen(陈单), Defu Cai(蔡德福), and Housheng Su(苏厚胜)
Chin. Phys. B, 2023, 32 (**9**): 098903. DOI: 10.1088/1674-1056/acd3e2

Synchronization in multilayer networks through different coupling mechanisms

Xiang Ling(凌翔), Bo Hua(华博), Ning Guo(郭宁), Kong-Jin Zhu(朱孔金), Jia-Jia Chen(陈佳佳), Chao-Yun Wu(吴超云), and Qing-Yi Hao(郝庆一)
Chin. Phys. B, 2022, 31 (**4**): 048901. DOI: 10.1088/1674-1056/ac2b19

Cascading failure in multilayer networks with dynamic dependency groups

Lei Jin(金磊), Xiaojuan Wang(王小娟), Yong Zhang(张勇), Jingwen You(由婧文)
Chin. Phys. B, 2018, 27 (**9**): 098901. DOI: 10.1088/1674-1056/27/9/098901

A multilayer network diffusion-based model for reviewer recommendation

Yiwei Huang(黄羿炜)[1,2], Shuqi Xu(徐舒琪)[3], Shimin Cai(蔡世民)[4,5,†], and Linyuan Lü(吕琳媛)[6,1,2,‡]

[1] *Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu* 611731, *China*
[2] *Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou* 313001, *China*
[3] *Institute of Dataspace, Hefei Comprehensive National Science Center, Hefei* 230088, *China*
[4] *Big Data Research Center, University of Electronic Science and Technology of China, Chengdu* 611731, *China*
[5] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu* 611731, *China*
[6] *School of Cyber Science and Technology, University of Science and Technology of China, Hefei* 230026, *China*

(Received 24 September 2023; revised manuscript received 22 December 2023; accepted manuscript online 22 December 2023)

With the rapid growth of manuscript submissions, finding eligible reviewers for every submission has become a heavy task. Recommender systems are powerful tools developed in computer science and information science to deal with this problem. However, most existing approaches resort to text mining techniques to match manuscripts with potential reviewers, which require high-quality textual information to perform well. In this paper, we propose a reviewer recommendation algorithm based on a network diffusion process on a scholar–paper multilayer network, with no requirement for textual information. The network incorporates the relationship of scholar–paper pairs, the collaboration among scholars, and the bibliographic coupling among papers. Experimental results show that our proposed algorithm outperforms other state-of-the-art recommendation methods that use graph random walk and matrix factorization and methods that use machine learning and natural language processing, with improvements of over 7.62% in recall, 5.66% in hit rate, and 47.53% in ranking score. Our work sheds light on the effectiveness of multilayer network diffusion-based methods in the reviewer recommendation problem, which will help to facilitate the peer-review process and promote information retrieval research in other practical scenes.

Keywords: reviewer recommendation, multilayer network, network diffusion model, recommender systems, complex networks

PACS: 89.75.Hc, 64.60.aq **DOI:** 10.1088/1674-1056/ad181d

1. Introduction

The publication of a scientific article often needs to undergo multiple steps, among which peer review plays an essential role.[1] Appropriate reviewers can offer valuable suggestions to authors for manuscript revision and assist editors in decision-making, thereby helping to improve the quality of the paper and facilitate the peer review process.[2] Despite editors being able to rely on their reviewer database and authors' recommendations to choose reviewers, finding an adequate number of qualified reviewers who are experienced in the research area of the submitted manuscripts is still tough, given the rapid growth of submissions.[3] To solve this problem, recommender systems aiming at reviewer recommendations are increasingly investigated and applied. Among them, research focusing on mining the relevance between reviewers and manuscripts via textual information accounts for a significant part,[2] in which methods based on natural language processing such as TF-IDF, LDA and LSI (see Subsection 2.1) find wide applications.[4] The text used for analysis mainly includes the submitted papers and the publications of reviewer candidates, specifically their titles, keywords, and abstracts. In addition, to enhance the accuracy of recommendations, some hybrid approaches combine text analysis with methods such as collaborative filtering, network models,[5] and rule-based systems.[6] In general, considerable efforts have been made in applying text analysis to match reviewers with manuscripts, while methods that do not require textual data have received limited discussion. However, the practical challenge of collecting complete textual information of reviewers, coupled with the occasional unavailability of full manuscripts, significantly impacts the accuracy of text analysis, calling for new methods that do not rely on textual data.

This paper proposes a reviewer recommendation algorithm that leverages a diffusion process on a constructed multilayer network. The contribution lies in two aspects: the novel multilayer-network model and the diffusion-based recommendation algorithm. Firstly, we construct a multilayer network that encompasses various potential relationships, including the reviewing relations between reviewers and submissions, the co-reviewing relations among reviewers, and the bibliographic coupling relations among submissions. These relations precisely capture the matching of research fields between reviewers and submissions, the research relatedness

†Corresponding author. E-mail: shimin.cai81@gmail.com
‡Corresponding author. E-mail: linyuan.lv@ustc.edu.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

among reviewers and the knowledge relatedness among submissions, respectively. Notably, our model does not require any textual information of reviewers and submissions, which reduces the difficulty in collecting high-quality data. To build the network, we extract review records from a conference that involves 70 reviewers, based on which we further collect 2804 related scholars and their published papers from Scopus (www.scopus.com). Due to the scarcity of large-scale empirical peer-review data,[7] we use authorship information to simulate reviewing relations, and use co-authorship connections among scholars to simulate co-reviewing relations.

Secondly, we propose an innovative diffusion-based recommendation algorithm. Specifically, we develop a new set of diffusion rules for the algorithm which explores the influence of different diffusion orders and proportions on the recommendation accuracy. Experimental results indicate that our proposed methods outperform the state-of-the-art methods that incorporate graph random walk and matrix factorization[8] and methods that incorporate machine learning and natural language processing by a wide margin, thus proving that considering collaboration relations among scholars and bibliographic coupling relations among papers help to improve the recommendation performance.

The remainder of this paper is organized as follows. Section 2 reviews the related studies focusing on the reviewer recommendation problem. Section 3 describes the newly built dataset and the construction of the two-layer network model. Section 4 elaborates on our network diffusion-based recommendation model, some benchmark methods, and evaluation metrics used in the study. Section 5 presents and analyzes the experiment results for the proposed and benchmark methods. Section 6 concludes this paper and discusses future works.

2. Related work

Automatically recommending reviewers for the submitted manuscripts can be traced back to Dumais and Nielsen's work[9] in 1992, which is also known as the reviewer assignment problem (RAP).[10] RAP can be divided into retrieval-based RAP (RRAP) and assignment-based RAP (ARAP),[11] in which the former treats each submitted paper as a query to retrieve the most relevant reviewers,[12] while the latter aims at achieving a balanced workload distribution among reviewers. Given that ARAP is only relevant when reviewer candidates are limited, whereas RRAP is more applicable in most reviewer recommendation scenarios, our study primarily focuses on RRAP. Based on the information used for analysis, RRAP can be further divided into text-based and non-text-based methods.[4,7]

2.1. Text-based methods

Commonly-used textual information in RRAP for manuscripts includes the full text (but sometimes it is unavailable), title, keywords, and abstract, as these directly reflect the content of the manuscript. As for reviewers, some online review systems require them to provide descriptions of their research fields. However, the quality of such information is often unstable and fails to completely reveal the reviewer's interests.[9] Therefore, building reviewers' representations through their publications (i.e., titles, keywords, and abstracts) has been an alternative.[13–15]

To address RRAP based on the aforementioned textual information, researchers commonly resort to natural language processing (NLP) techniques. This usually involves representing manuscripts and reviewers as vectors or matrices, which are then used as input to compute the matching degree between them. The generated representation falls into two levels, namely word-level and topic-level.[16]

At the word level, there are three main approaches:[7] the term frequency-inverse document frequency (TF-IDF), language model (LM), and keyword matching. TF-IDF is a widely used word-level NLP method employed in the vector space model (VSM), where all available textual information of the submitted papers and reviewers' publications are represented as weighted vectors.[13,17,18] Then the relationship between a submission and a reviewer is assessed by measuring the similarity between their respective vectors using metrics like cosine similarity. Language models analyze reviewers' publications to represent each reviewer as a distribution over words in the vocabulary. Then the relevance between reviewers and a given submission is measured based on the probability of each word in the submission appearing in the reviewer's distribution. Different from language models targeting the co-occurrence of words in submissions and reviewers' publications, keyword matching methods focus on keyword co-occurrence. Here the keywords can be predefined by conference chairs,[19] provided by reviewers, or extracted from submissions and reviewers' papers.[20] Since exact keyword matching may fail to identify two documents that are related to each other yet have no common keywords, Yordan Kalmukov[19] built a taxonomy of keywords that reveals the semantic relationships between different keywords, making it possible to compute the similarity between a submission and a reviewer even without shared keywords. Tayal *et al.*[10] created fuzzy sets of reviewers and submissions, calculating fuzzy equality rather than the number of common keywords.

Compared with word-level analysis which often brings high-dimensional representations of papers and reviewers, topic models represent the submitted papers and reviewers' publications as a topic distribution where each topic is represented as a distribution over words that are trained from the

corpus. Dumais and Nielsen[9] used the latent semantic indexing (LSI) model that employs singular-value decomposition (SVD) to obtain a relevance matrix between papers and topics. The drawback of LSI is that its topic distribution output lacks a probabilistic interpretation. By contrast, the latent dirichlet allocation model[21] (LDA) assigns probabilities to individual words in each topic, making it easier to understand the semantic meaning of the identified topics. In addition, the generative process of LDA allows it to handle unknown text. As a result, LDA and its variants have been popular in RRAP. Mimno and McCallum[22] proposed the author–persona–topic (APT) model that allows each reviewer to have multiple topic distributions to represent their interdisciplinary expertise. Kou *et al.*[11] applied the author–topic model (ATM)[23] to represent reviewers and proposed a weighted coverage-based method that is capable of assigning a reviewer group with comprehensive expertise to an interdisciplinary paper. To recommend reviewers for interdisciplinary submissions, Jin *et al.*[24] proposed the author–subject–topic (AST) model by introducing a supervised subject layer to model the knowledge structure of reviewers more precisely. Peng *et al.*[25] presented a time-aware topic-based reviewer assignment model that captures reviewers' changes in research interest with time.

In recent years, neural network techniques have also been applied to reviewer assignments to learn more effective representations of reviewers and submissions, such as word2vec,[12,26] bidirectional encoder representation from transformers (BERT)[4] and gated recurrent unit (GRU).[16]

2.2. Non-text-based methods

Text-based methods can be very expensive regarding computational costs, especially when applied to a large dataset. Fortunately, alternative non-text-based methods that do not pose significant computational challenges are available, including network-based, collaborative filtering, and rule-based methods. Note that most of them still need text analysis to process the data.

Network-based methods leverage the topology structure of various relationships among reviewers and manuscripts to recommend reviewers. Reviewer networks and reviewer–manuscript networks are the two basic models. For the former, Yin *et al.*[27] employed co-authorship relations to connect reviewers. Jin *et al.*[28] built a citation network among reviewers, and introduced a topical PageRank algorithm[29] to evaluate reviewers' authority in the topics of the submitted paper. To narrow the range of potential reviewers when they are not predetermined (usually in the journal peer-review process), Rodriguez and Bollen[30] regarded the authors of a manuscript's references as its potential reviewers and built a co-authorship network among them, using a particle swarm algorithm on it to identify appropriate reviewers. As for the reviewer–manuscript network model, reviewers are connected to manuscripts according to their preferences or their relevance of expertise. Goldsmith and Sloan[31] created a reviewer–conference submission network, where the weight of links is determined based on reviewers' ratings of their willingness to review the submissions before assignment. Then they used a network flow algorithm through this network to assign submissions to reviewers. But when the information of reviewers' preferences is unavailable, it is necessary to connect reviewers and manuscripts based on their textual information. Liu *et al.*[5] used semantic similarity scores between reviewers and manuscripts obtained through LDA analysis as the weight of each link and employed a random walk with restart (RWR) model on the network[32] to balance the expertise and authority of recommended reviewers. Except for the two kinds of basic models, researchers also built networks with more diverse structures, such as three-layer networks[33,34] and multiple networks.[35] Based on the constructed network, one can also utilize node ranking methods[36,37] or topology identification methods[38] to make recommendations.

Collaborative filtering (CF) methods[39] generate recommendations by leveraging the information of reviewers' preferences on submitted papers. These methods are commonly used in the review process concerning conference papers, in which there is usually a bidding process before assignment that requires reviewers to rate the submitted papers according to their reviewing willingness. Since it is impractical to require reviewers to rate all papers, the rating matrix tends to be sparse. Collaborative filtering methods aim at utilizing existing ratings to infer the unspecified ones, which mainly include two categories: memory-based CF and model-based CF.[40] Memory-based CF predicts reviewers' preferences by computing the similarity among reviewers, under the basic assumption that reviewers sharing similar ratings on some papers are likely to rate similarly on others.[41,42] And model-based CF focuses on learning a predictive model from the rating data. For example, Conry[43] utilized latent factor models to factorize the reviewer–paper rating matrix into two matrices containing the latent factor representation of reviewers and papers, whose dot product estimates the unknown ratings.

Rule-based methods usually represent the knowledge (such as procedures and constraints) of the peer-review process as a set of IF-THEN rules. For example, if a reviewer and a submission's authors are from the same institution, then there may exist conflicts of interest. This type of constraint can not be directly handled using mathematical methods (such as optimization algorithms), but can be effectively addressed by encoding it as a set of rules that can be checked and enforced. As such, Mauro *et al.*[6] presented a rule-based system for the review of conference papers, in which the authors designed a set of rules to achieve a solution with feasibility (rules such

as the maximum number of assigned papers of per reviewer), effectiveness (rules measuring the confidence degree of the assignment solution), and fairness (rules for avoiding conflicts of interest between reviewers and authors).

3. Data and network construction

3.1. Dataset

To the best of our knowledge, there is no public dataset that records the complete paper reviewing process from journals or conferences, and the large-scale well-documented relationship between reviewers and papers is hardly available. Thus we use the relationship between scholars and their published papers to simulate the peer review relations. Such a simulation is reasonable because a published paper directly reflects its author's research fields, and the scholar who published a paper is certainly capable of reviewing it.

To narrow our focus on a peer review scenario, we employ the 70 reviewers in NetSci-X 2018 (netscisociety.net/events/netsci) conference as the initial data, then collect their coauthor lists from the Scopus database through the API Key (dev.elsevier.com) and a Python interface,[44] as well as papers of all reviewers and their coauthors. To ensure the feasibility of recommendations, we only preserve papers with no less than two authors. Finally, we get a dataset involving 2804 scholars and 24197 papers, with publication dates ranging from March 1st, 1973 to February 1st, 2021. The scholar information includes the list of published papers and coauthors, while the paper information includes its authors and list of references.

To perform a comparison with text-based methods, we also search the textual information of each paper in our dataset from the Scopus database, and finally we collect the title and abstract of 23899 papers, which will be used as the input corpus for the text-based model training (see Subsection 4.2.1).

3.2. Two-layer network construction

According to the collected dataset, we extract three kinds of relationships among scholars and papers, based on which we construct a scholar–paper two-layer network. In the scholar layer, we link scholars based on their co-authorship relations. In other words, if two scholars coauthor at least one paper in the dataset, there exists a link between them. To construct the paper layer, we analyze the reference list of each paper and connect papers according to their bibliographic coupling relations. Namely, there exists a link between two papers if they have at least one co-citing item, and the link weight denotes the number of coupling references. Given that the number of this type of link is enormous, we only preserve links with a weight larger than 4 in the constructed network. And the main results are robust with respect to variations of this threshold. Finally, we use the authorship relations between papers and their authors to connect two layers. This means the link connecting an author with a paper denotes that this author wrote the paper. Figure 1 shows a simplified illustration of the constructed two-layer network, which is made up of three different sub-networks: a co-authorship network in the scholar layer, a bibliographic coupling network in the paper layer, and a publishing network between two layers. Note that when we use real peer review data to construct such a network, things will be slightly different, where authorship relationships need to be replaced with reviewing relationships, and co-authorship relationships need to be changed to co-reviewing relationships (see details in Appendix A).

Table 1. The statistical properties of the two-layer network.

Networks	Nodes	Links	$\langle k \rangle$	
			$\langle k_\alpha \rangle$	$\langle k_i \rangle$
Publishing network	27,001	73,637	26.26	3.04
Co-authorship network	2,804	24,072	17.17	–
Bibliographic coupling network	24,197	580,646	–	47.99

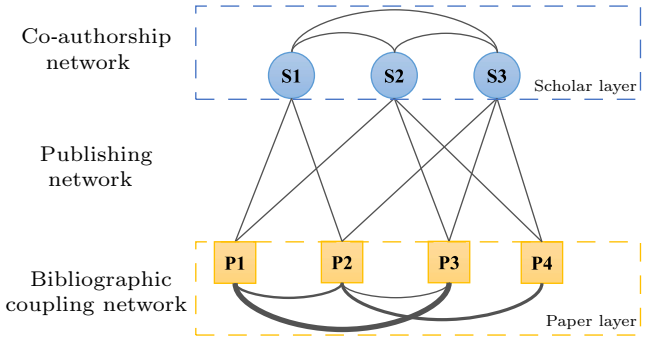


Fig. 1. Illustration of the scholar–paper two-layer network. There are two layers composed of scholar nodes and paper nodes, respectively, and three types of links, which represent publishing, co-authorship, and bibliographic coupling relations, respectively. In the bibliographic coupling network, the thickness of each link is proportional to its weight, which is determined by the number of common references between two papers. Links in the publishing network and co-authorship network are unweighted.

Mathematically, we denote the network representation as $\mathcal{G}(P, S, E_{PS}, E_{SS}, E_{PP})$, where P represents the set of n papers, S represents the set of m scholars. E_{PS}, E_{SS} and E_{PP} represent the set of links in the publishing network, co-authorship network and bibliographic coupling network, respectively, whose mathematical descriptions are as follows.

As for the three sub-networks, the publishing network is represented as an adjacency matrix $\boldsymbol{A}(P, S, E_{PS})$, in which $A_{i\alpha} = 1$ if scholar α is the author of paper i, otherwise $A_{i\alpha} = 0$. In this network, the degree of paper i (k_i^A) equals the number of its authors, and the degree of scholar α (k_α^A) equals the number of papers published by the scholar. Similarly, the co-authorship network is also represented as an adjacency matrix $\boldsymbol{C}(S, E_{SS})$. If scholar α and β have published at least one paper as coauthors, then $C_{\alpha\beta} = 1$, or else $C_{\alpha\beta} = 0$. In this network, the degree of scholar α (k_α^C) equals the number of

the scholar's coauthors. Different from the above two sub-networks, the bibliographic coupling network is a weighted network, which is denoted as $B(P, E_{PP})$, where $B_{ij} = w$ denotes that papers i and j have w common references. We refer to w_i^B as the sum of weights of the links related to paper i in the bibliographic coupling network. Table 1 shows the statistical properties of the three sub-networks. Figure 2 illustrates the degree distributions of nodes in three sub-networks, where we can observe an approximate power-law distribution in the bibliographic coupling network.

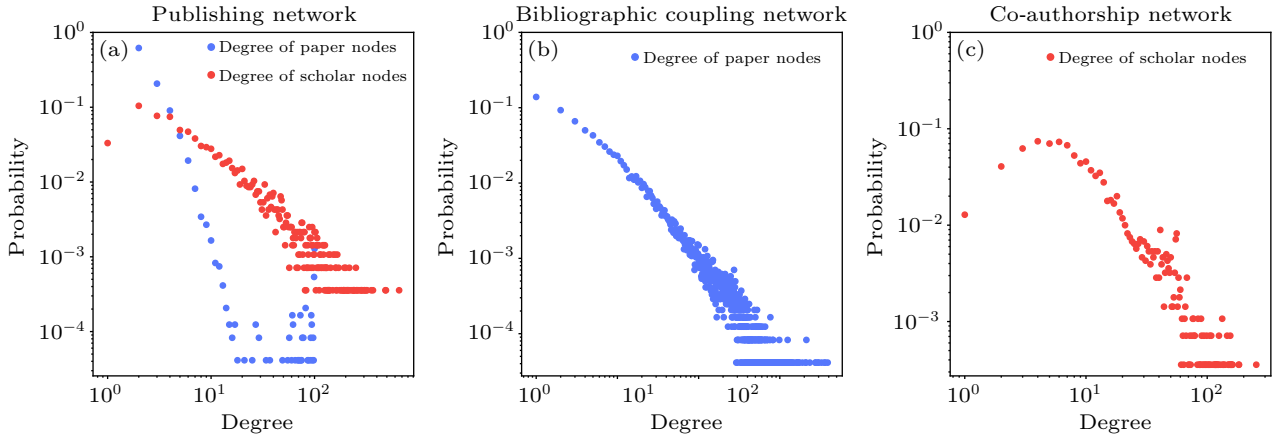


Fig. 2. Degree distributions of paper nodes and scholar nodes for the publishing network (a), bibliographic coupling network (b), and co-authorship network (c).

4. Recommendation methods

4.1. Recommendation based on the two-layer network

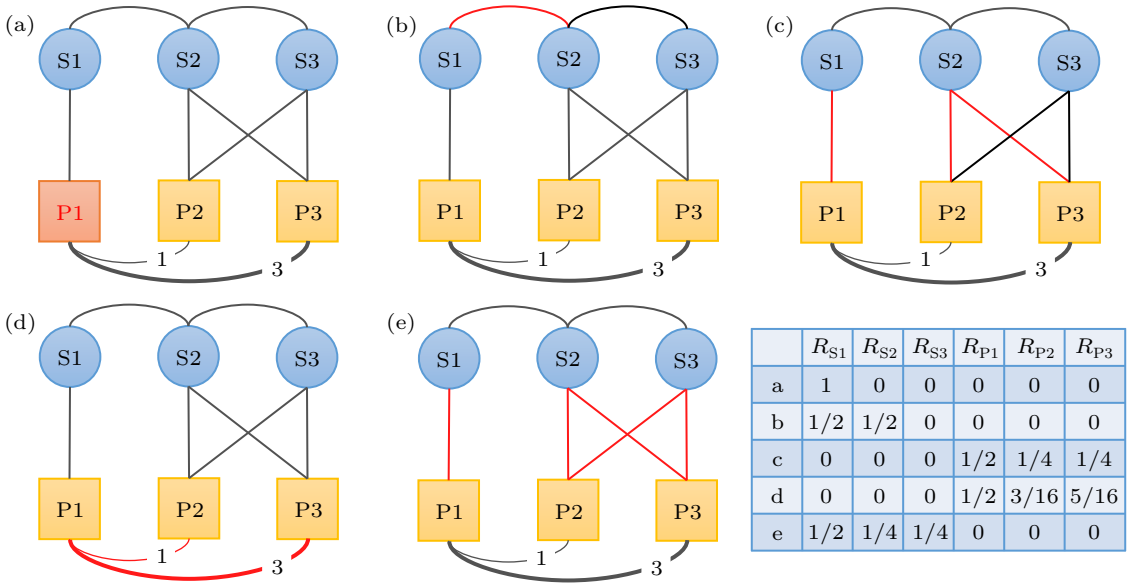
Diffusion-based algorithms are commonly-used methods to implement recommendations or study the dynamics of information diffusion[45,46] in networks, such as random walk,[5] mass diffusion[47] and heat conduction,[48–51] whose diffusion processes are derived from physical phenomena. Random walk models are often utilized to describe the stochastic motion of particles in physical systems. Mass diffusion models illustrate the diffusion process of particles from regions of higher concentration to regions of lower concentration, following the principle of the conservation of matter. Heat conduction models describe the transfer of heat energy from regions of higher temperature to regions of lower temperature.

Among these diffusion-based algorithms, mass diffusion[47] is recognized as one of the most efficient and concise methods which assumes that nodes in the network are assigned a kind of resources, and resources can diffuse from these nodes to other nodes along network links. The standard mass diffusion algorithm is applied on a bipartite network, where there exists only one type of link.[52–54] However, when we employ this algorithm on our two-layer network that includes three types of links (scholar–scholar, paper–paper, and scholar–paper), two factors need to be determined first: (i) diffusion order: whether resources diffuse within the same layer (i.e., scholar–scholar) first or across the opposite layer (i.e., scholar–paper) first? (ii) diffusion proportion:[55] what proportions of resources should be distributed between scholar pairs and between paper pairs?

Specifically, at the very beginning, we assign the initial resource to the authors of a target paper, who are simulated as the reviewers. Subsequently, there will be two choices to start the diffusion: distribute-first or diffuse-first. As for the distribute-first way, the initial resource of each scholar node will first be distributed to its neighboring scholar nodes with a proportion of $p_2 \in [0,1]$, while the remaining $(1-p_2)$ proportion of resources is reserved. Then, the resources in this scholar layer begin to diffuse to the paper layer. In contrast, in the diffuse-first method, initial resources will first transfer from scholar nodes to their connected paper nodes. Then, $p_1 \in [0,1]$ proportion of resources of these paper nodes begin to be distributed to their neighboring paper nodes. Parameters p_1 and p_2 define the proportion of resources involved in the redistribution on the paper and scholar layer, respectively. When $p_1 = 0$ and $p_2 = 0$, the method reduces to standard mass diffusion on the scholar–paper bipartite network.

As illustrated in Fig. 3, suppose that paper P1 is the target paper, which requires recommended reviewers. The diffusion process of the distribute-first recommendation for P1 involves five steps. Step (a): initial resources assignment. All scholars connected with P1 are assigned one unit of initial resources, and the others are assigned 0. Step (b): resource redistribution among scholars. Each scholar equally transfers p_2 proportion of resources to their coauthors in the co-authorship network, while keeping a proportion of $(1-p_2)$ for themselves. After that, all scholars update their resources by adding the reserved and the obtained resources together. Step (c): resource diffusion from scholars to papers. Scholars transfer their resources to their connected papers in the publishing network. Then all nodes update their resources. Step (d): resource redistribution among papers. Each paper distributes p_1 proportion of resources to their connected papers in the bibliographic cou-

pling network, proportionally based on the link weights, while keeping a proportion of $(1-p_1)$ for themselves. Then, each paper updates its resources. Step (e): the diffusion from papers to scholars, which is similar to step (c). After that, each scholar updates the final resources, which are considered as the recommendation score. Here the score of scholar S1, S2 and S3 is $1/2$, $1/4$ and $1/4$, respectively. As to the diffuse-first recommendation, the order of the steps will become (a), (c), (d), (e), (b), and the final resources of scholar S1, S2 and S3 will be $5/16$, $1/2$ and $3/16$, respectively. For this diffusion process on the reviewer–manuscript two-layer network, see Appendix A for an example.



	R_{S1}	R_{S2}	R_{S3}	R_{P1}	R_{P2}	R_{P3}
a	1	0	0	0	0	0
b	1/2	1/2	0	0	0	0
c	0	0	0	1/2	1/4	1/4
d	0	0	0	1/2	3/16	5/16
e	1/2	1/4	1/4	0	0	0

Fig. 3. A simple example of the distribute-first recommendation with $p_1 = p_2 = 1/2$. Paper P1 is assumed to be the target paper that needs to be recommended by reviewers. In each step, links in red are those involved in the diffusion process. See details of each step in the main text. The table records the resource value of each node after each step.

Mathematically, we denote the initial resources assigned to scholar α as R_α, and the final resources scholar γ received as R'_γ, which is also considered as the recommendation score of scholar γ. Here we have the equations between R_α and R'_γ below (see the detailed derivation in Appendix B).

In the way of distribute-first,

$$\begin{aligned} R'_\gamma = \Bigg[& (1-p_1)(1-p_2)\sum_{i=1}^{n}\sum_{\alpha=1}^{m}\frac{A_{i\gamma}A_{i\alpha}}{k_i^A k_\alpha^A} \\ & +(1-p_1)p_2\sum_{i=1}^{n}\sum_{\alpha=1}^{m}\sum_{\beta=1}^{m}\frac{A_{i\gamma}A_{i\beta}C_{\alpha\beta}}{k_i^A k_\beta^A k_\alpha^C} \\ & +p_1(1-p_2)\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{m}\frac{A_{j\gamma}B_{ij}A_{i\alpha}}{k_j^A w_i^B k_\alpha^A} \\ & +p_1p_2\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{m}\sum_{\beta=1}^{m}\frac{A_{j\gamma}B_{ij}A_{i\beta}C_{\alpha\beta}}{k_j^A w_i^B k_\beta^A k_\alpha^C}\Bigg] R_\alpha. \end{aligned} \tag{1}$$

In the way of diffuse-first,

$$\begin{aligned} R'_\gamma = \Bigg[& (1-p_1)(1-p_2)\sum_{i=1}^{n}\sum_{\alpha=1}^{m}\frac{A_{i\gamma}A_{i\alpha}}{k_i^A k_\alpha^A} \\ & +(1-p_1)p_2\sum_{i=1}^{n}\sum_{\alpha=1}^{m}\sum_{\beta=1}^{m}\frac{C_{\beta\gamma}A_{i\beta}A_{i\alpha}}{k_\beta^C k_i^A k_\alpha^A} \\ & +p_1(1-p_2)\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{m}\frac{A_{j\gamma}B_{ij}A_{i\alpha}}{k_j^A w_i^B k_\alpha^A} \\ & +p_1p_2\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{m}\sum_{\beta=1}^{m}\frac{C_{\beta\gamma}A_{j\beta}B_{ij}A_{i\alpha}}{k_\beta^C k_j^A w_i^B k_\alpha^A}\Bigg] R_\alpha. \end{aligned} \tag{2}$$

According to the recommendation scores, we rank scholars that are not connected with the target paper in descending order, and the top L of them would be the most eligible reviewers for this paper.

4.2. Benchmark methods

4.2.1. Text-based methods

In this study, we construct the documents for target papers by concatenating their titles and abstracts, and establish the documents for reviewers by concatenating the titles and abstracts of their publications.[14,15,56] We then input these documents into text-based models and generate a representation for each of them, based on which we measure the semantic similarity between target papers and reviewers. Here we consider four representative models, which use word-level information, topic-level information, external information and context information, respectively.

Vector space model (VSM) VSM extracts document features for target papers and reviewers from word-level information, where words are weighted by TF-IDF scores. TF-IDF is a statistical method that evaluates the relevance of a word to a document in the whole document corpus. The term frequency (TF) measures how often a word appears in a certain docu-

ment, while the inverse document frequency (IDF) measures how important a word is across the entire corpus. Words that appear frequently in a specific document but rarely in other documents are considered more important for that document, and thus have higher TF-IDF scores. The TF-IDF score of word i in a document α is defined as the product of its TF and IDF

$$S(i,\alpha) = TF_{i,\alpha} \times \log \frac{|D|}{|\{d|d \in D, i \in d\}|}, \tag{3}$$

where D is the set of documents in the corpus, and the denominator is the number of documents containing the word i. By computing the TF-IDF score of all unique words in the vocabulary, we can map the document of the target paper and the reviewer into the same vector space, and can calculate their cosine similarity, which serves as the reviewer's recommendation score.

Latent dirichlet allocation model (LDA)[21] LDA is an unsupervised machine learning algorithm, which is used to model the underlying topics that are present in a large collection of text documents based on word co-occurrences. Each topic is represented in the form of a word probability distribution. Then the documents of target papers and reviewers can be represented as probability distributions over topics. Thus we can obtain a reviewer's recommendation score for a target paper by calculating the cosine similarity between the topic probability distributions. As for the number of topics ($n_{\rm t}$), we test a continuous set of values and use the perplexity and topic coherence[57] metric to determine $n_{\rm t} = 50$ for a trade-off between predictive performance and interpretability.

Random walk with restart model based on LDA analysis (LDA-RWR)[5] LDA-RWR is a hybrid model that integrates text analysis and network propagation process to generate recommendations. For each target paper, we build a network composed of this paper and all reviewers, where the link weights between the reviewers and the target paper are the topic similarity measured by the LDA model, and the link weights among reviewers are the number of co-authored papers. We set the target paper as the starting point of a random walk with a restart process on the network. The probability scores of reviewers in the stationary distribution of the process are their recommendation scores. In the random walk process, the damping factor η ($\eta \in [0,1)$) determines the probability of jumping back to the target paper. In the experiment, we search for the optimal η with an interval of 0.1 and find $\eta = 0.1$ corresponds to the highest recommendation accuracy.

Word2vec model with soft cosine measure (W2V-SCM) Word2vec is a neural network technique that learns the linguistic contexts of words. It embeds words into a vector space where words that share similar contexts in the corpus are located close to each other. Based on the representation of words, we use soft cosine measure (SCM)[58] to calculate the semantic similarity between the documents of target papers and reviewers. Specifically, suppose that there are $N_{\rm w}$ words in the vocabulary, the documents of target paper i and reviewer α can be represented as $N_{\rm w}$-dimensional vectors

$$\boldsymbol{V}_i = [n_{i1}, n_{i2}, \ldots, n_{iN_{\rm w}}], \quad \boldsymbol{V}_\alpha = [n_{\alpha 1}, n_{\alpha 2}, \ldots, n_{\alpha N_{\rm w}}], \tag{4}$$

where the elements correspond to the frequencies of words. The soft cosine similarity between V_i and V_α is defined as the recommendation score of reviewer α for target paper i

$$S(i,\alpha) = \frac{\sum_{a=1}^{N_{\rm w}} \sum_{b=1}^{N_{\rm w}} s_{ab} n_{ia} n_{\alpha b}}{\sqrt{\sum_{a=1}^{N_{\rm w}} \sum_{b=1}^{N_{\rm w}} s_{ab} n_{ia} n_{ib}} \sqrt{\sum_{a=1}^{N_{\rm w}} \sum_{b=1}^{N_{\rm w}} s_{ab} n_{\alpha a} n_{\alpha b}}}, \tag{5}$$

where s_{ab} is the conventional cosine similarity between the embeddings of words a and b. In the experiment, we test different embedding dimensions and find that the optimal one is 200.

4.2.2. Non-text-based methods

User-based collaborative filtering (UCF) Collaborative filtering (CF) is a basic approach in recommender systems, which can be divided into memory-based and model-based collaborative filtering. Within the memory-based approach, user-based collaborative filtering (UCF) and item-based collaborative filtering (ICF)[40] are the most widely-used methods. The main idea behind UCF is that similar users share similar tastes in items. In the scenario of recommending reviewers, this approach will recommend scholars who are connected with papers similar to the target paper as reviewers. Its implementation includes two steps: (i) calculating the similarity between the target paper and other papers. Considering the paper nodes are involved in two sub-networks (i.e., the publishing network and the bibliographic coupling network), we compute the similarity among papers separately in these two sub-networks. In the publishing network, the cosine similarity between any two papers i and j can be calculated as in Ref. [59]

$$\mathrm{sim}_A(i,j) = \frac{|\Gamma_A(i) \cap \Gamma_A(j)|}{\sqrt{|\Gamma_A(i)| \times |\Gamma_A(j)|}}, \tag{6}$$

where $\Gamma_A(i)$ refers to the set of scholars that connect with paper i in the publishing network. In the bibliographic coupling network, two factors need to be considered when we measure the similarity of two papers: one is the link weight between them, and the other is the number of their common neighbors. Binesh *et al.*[60] pointed out that both factors are all important and proposed a modified cosine similarity to combine them. In this method, the similarity between two papers i and j is

$$\mathrm{sim}_B(i,j) = \frac{\boldsymbol{b}_i \cdot \boldsymbol{b}_j}{|\boldsymbol{b}_i| \, |\boldsymbol{b}_j|},$$

$$\boldsymbol{b}_i = [B_{i1}, B_{i2}, \ldots, B_{ii-1}, w_i^B, B_{ii+1}, \ldots, B_{in}], \tag{7}$$

where $\boldsymbol{b}_i$ is an n-dimensional vector, in which the nonzero elements represent the weight of links between paper i and other papers, and the i-th element is replaced by the sum of these nonzero values (w_i^B).[60] Such modification has been proved to perform better compared with the original cosine similarity in recommendation,[60] and we also obtain a consistent conclusion in our study. (ii) For each target paper, the recommendation score of scholar α for paper i is defined as in Ref. [61]

$$\begin{aligned} S(i,\alpha) = {} & \lambda_{\mathrm{UCF}} \sum_{j \in \Gamma_A(\alpha)} \mathrm{sim}_A(i,j) \\ & + (1-\lambda_{\mathrm{UCF}}) \sum_{j \in \Gamma_A(\alpha)} \mathrm{sim}_B(i,j), \end{aligned} \tag{8}$$

where $\Gamma_A(\alpha)$ refers to the set of papers connected with scholar α in the publishing network, and λ_{UCF} is used to adjust the two types of similarity, $0 \le \lambda_{\mathrm{UCF}} \le 1$.

Item-based collaborative filtering (ICF) Unlike UCF, the ICF approach will recommend scholars who exhibit similarities to the scholars connected to the target paper as reviewers. Its implementation also consists of two steps: (i) computing the similarity of scholar candidates to the scholars connected to the target paper. Similarly, we compute the similarity among scholars through the two sub-networks. In the publishing network, the cosine similarity between any two scholars α and β can be calculated as follows:[59]

$$\mathrm{sim}_A(\alpha,\beta) = \frac{|\Gamma_A(\alpha) \cap \Gamma_A(\beta)|}{\sqrt{|\Gamma_A(\alpha)| \times |\Gamma_A(\beta)|}}, \tag{9}$$

where $\Gamma_A(\alpha)$ refers to the set of papers that connect with scholar α in the publishing network. In the unweighted co-authorship network, the modified cosine similarity between any two scholars α and β can be calculated as in Ref. [60]

$$\begin{aligned} & \mathrm{sim}_C(\alpha,\beta) = \frac{\boldsymbol{c}_\alpha \cdot \boldsymbol{c}_\beta}{|\boldsymbol{c}_\alpha|\,|\boldsymbol{c}_\beta|}, \\ & \boldsymbol{c}_\alpha = [C_{\alpha 1}, C_{\alpha 2}, \ldots, C_{\alpha\alpha-1}, 1, C_{\alpha\alpha+1}, \ldots, C_{\alpha m}], \end{aligned} \tag{10}$$

where $\boldsymbol{c}_\alpha$ is an m-dimensional vector, in which the elements represent whether there is a link between scholar α and other scholars, and the α-th element is replaced with 1.[60] Similarly, the performance of ICF also improves with such modifications in our experiments. (ii) Recommending scholars that are similar to the scholars connected to the target paper as potential reviewers. The recommendation score of scholar α for paper i is defined as in Ref. [61]

$$\begin{aligned} S(i,\alpha) = {} & \lambda_{\mathrm{ICF}} \sum_{\beta \in \Gamma_A(i)} \mathrm{sim}_A(\alpha,\beta) \\ & + (1-\lambda_{\mathrm{ICF}}) \sum_{\beta \in \Gamma_A(i)} \mathrm{sim}_C(\alpha,\beta), \end{aligned} \tag{11}$$

where $\Gamma_A(i)$ refers to the set of scholars connected with paper i in the publishing network, and λ_{ICF} is used to adjust similarity of these two types, $0 \le \lambda_{\mathrm{ICF}} \le 1$.

High-order proximity augmented recommendation (HOP-Rec)[8] HOP-Rec is a state-of-the-art recommendation method incorporating graph-based and matrix factorization-based approaches. It utilizes random walks on the user–item graph to explore the high-order interactions among nodes to overcome the sparsity of bipartite interactions. According to this approach and our scholar–paper bipartite network, the high-order relevance between papers and scholars is computed through a random walk process (see Ref. [62] for details), with walk length K set to 2 as a trade-off between performance and complexity. After constructing the matrix factorization model trained with the Bayesian personalized ranking learning (BPR) method, the high-order relevance matrix is decomposed to learn the latent factors of scholars and papers. The recommendation score of a scholar for a target paper is estimated as the dot product of the corresponding factors. See the implementation details in Ref. [62]. Here we apply a grid search for the best combination of hyper-parameters, specifically, the dimension of latent factors d is 128 and the learning rate r is 0.01.

Popularity-based recommendation (POP) We define scholars with more linked papers and scholars as popular ones. Specifically, we introduce a parameter λ_{POP} to linearly combine scholars' degrees in the publishing network and in the co-authorship network, which are normalized by the maximum degree of scholars in each network. The recommendation score of scholar α is computed as

$$S(\alpha) = \lambda_{\mathrm{POP}} \frac{k_\alpha^A}{k_{\max}^A} + (1-\lambda_{\mathrm{POP}}) \frac{k_\alpha^C}{k_{\max}^C}, \tag{12}$$

where $k_{\max}^A$ and $k_{\max}^C$ represent the maximum degree among all scholars in the publishing network and co-authorship network, respectively.

Random recommendation (Random) For each target paper, we randomly generate a list of recommended scholars, with both the scholars and their rankings being completely randomized.

4.3. Evaluation

To evaluate the proposed recommendation methods, we delete 10% of the links in the publishing network and test the methods' ability to predict these missing links. Specifically, we randomly remove 7363 scholar–paper links related to papers published between 2017 and 2021 in the publishing network, forming the testing set. The remaining 90% (66274 links) compose the training set, based on which we construct the experimental network. Then we combine this new publishing network with the coauthor list of scholars and reference list of papers to obtain the final two-layer network for conducting

experiments. Note that when removing links from the network to create the testing set, precautions are taken to avoid the cold-start problem by not removing links that would result in isolated nodes. We also make experiments on the 95%/5% and 85%/15% training/testing set splittings, and the results are consistent with those of our 90%/10% splitting (see details in Appendix C).

Considering that a paper only has 3.04 authors on average (see Table 1), the value of precision will be extremely low when the recommendation list length is long, with false positive samples (i.e., non-authors recommended) occupying most of it. Since we focus on the model's ability to identify positive samples (i.e., authors), we employ recall, hit rate,[63] and ranking score[47] as evaluation metrics to characterize the performance of the proposed methods and benchmark methods. For a better explanation, here we refer to the scholars connected with the target paper in the testing set as relevant scholars.

Recall Recall represents the probability that the relevant scholars are recommended to a target paper.[40] The recall of paper i is defined as

$$\text{Recall}_i = \frac{d_i}{D_i}, \tag{13}$$

where d_i means the number of recommended relevant scholars for paper i, and D_i means the total number of all relevant scholars for paper i. We can obtain the recall of a method by averaging the recall values of all target papers. Obviously, higher recall indicates better performance.

Hit rate (HR) If at least one relevant scholar is recommended to a target paper, we will consider this paper as a hit paper. HR quantifies the fraction of hit papers among all target papers, which can be computed as in Ref. [63]

$$\text{HR} = \frac{H}{N}, \tag{14}$$

where H means the number of hit papers, and N means the number of all target papers. High HR represents high accuracy of recommendations.

Ranking score (RS) The relevant scholars' ranking is also an important measure of the recommendation performance.[47] For a target paper i, if its relevant scholar α ranks $r_{i\alpha}$ among all scholars that are not connected with it in the training set, then the RS of α for i is computed as

$$\text{RS}_{i\alpha} = \frac{r_{i\alpha}}{M_i}, \tag{15}$$

where M_i represents the number of scholars who are not connected with paper i in the training set. We obtain the mean RS by averaging over all target papers in the testing set. A good recommendation method is expected to rank relevant scholars as high as possible, which means the RS of relevant scholars should be relatively low.

5. Results

5.1. Different diffusion orders and proportions

To analyze the impact of diffusion orders and proportions on the recommendation performance, we implement recommendations through the diffusion-first way and distribute-first way, respectively, and adjust the proportion values p_1 and p_2 with an interval of 0.05. For each combination of p_1 and p_2, we demonstrate the average results of 10 experiments in Fig. 4.

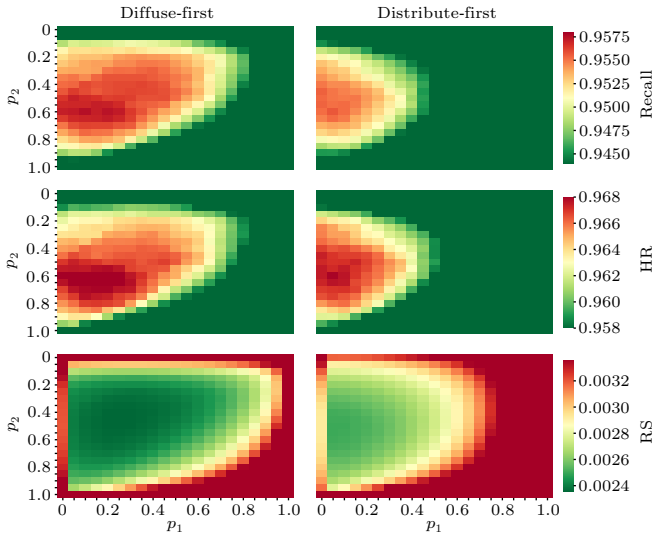


Fig. 4. The recommendation performance of the diffuse-first method and distribute-first method under different combinations of p_1 and p_2. Here all the results are averaged over 10 experiments, with the recommendation list length $L = 20$. The results demonstrate that the diffuse-first recommendation performs better than the distribute-first recommendation, as shown by the larger red areas in the left column. The optimal accuracy values in terms of the three evaluation metrics and the corresponding (p_1, p_2) are reported in Table 2.

The results indicate that the diffuse-first recommendation outperforms the distribute-first recommendation in terms of all the three evaluation metrics. Specifically, the diffuse-first recommendation achieves an average recall value that is 1.73% higher than that of the distribute-first method, a 1.10% higher HR, and a 12.48% lower RS (where a lower value indicates better performance). In addition, as shown in Fig. 4, across the different parameter combinations represented by the boxes, the diffuse-first recommendation method surpasses the distribute-first method in terms of recall in 95.01% of those combinations, HR in 88.44% of the cases, and RS in 86.17% of the cases.

As for the optimal performance (see Table 2), the highest recall achieved by the diffuse-first recommendation is 0.9575 (when $p_1 = 0.10$, $p_2 = 0.40$), while that for the distribute-first way is 0.9557 (when $p_1 = 0.05$, $p_2 = 0.40$); the highest HR values reached by the two methods are comparable; in terms of ranking score, the diffuse-first recommendation reaches 5.87% lower than the distribute-first method. In general, the inferior performance of the distribute-first method indicates that prioritizing resource redistribution among schol-

ars as the initial step will make the resources more dispersed, which is detrimental to the recommendation process. Besides, we find that the optimal accuracy values are achieved when both $p_1 \neq 0$ and $p_2 \neq 0$, suggesting that incorporating co-authorship and bibliographic coupling relations in the recommendation process enhance the reviewer recommendation performance. Furthermore, the optimal accuracy is achieved when $p_1 < p_2$ implies the greater importance of resource redistribution among scholars compared to papers, emphasizing the significance of allocating a higher proportion of resources to scholar redistribution for enhancing the recommendation system's performance.

Table 2. The optimal performance of the studied recommendation methods under three metrics, combining with the corresponding parameters ($L = 20$ for recall and HR). The bold values correspond to the best performance among all methods.

Methods	Recall	HR	RS
Diffuse-first	**0.9575** ($p_1 = 0.10, p_2 = 0.40$)	**0.9685** ($p_1 = 0.20, p_2 = 0.35$)	**0.0024** ($p_1 = 0.25, p_2 = 0.55$)
Distribute-first	0.9557 ($p_1 = 0.05, p_2 = 0.40$)	0.9679 ($p_1 = 0.05, p_2 = 0.40$)	0.0025 ($p_1 = 0.05, p_2 = 0.50$)
VSM	0.7753	0.8201	0.0148
LDA	0.3822	0.4351	0.0404
LDA-RWR	0.4015	0.4705	0.0346
W2V-SCM	0.6348	0.6894	0.0265
UCF	0.8878 ($\lambda_{\mathrm{UCF}} = 0.24$)	0.9160 ($\lambda_{\mathrm{UCF}} = 0.24$)	0.0213 ($\lambda_{\mathrm{UCF}} = 0.26$)
ICF	0.8879 ($\lambda_{\mathrm{ICF}} = 0.85$)	0.9113 ($\lambda_{\mathrm{ICF}} = 0.85$)	0.0048 ($\lambda_{\mathrm{ICF}} = 0.88$)
HOP-Rec	0.8290	0.8659	0.0336
POP	0.0749 ($\lambda_{\mathrm{POP}} = 0.79$)	0.1094 ($\lambda_{\mathrm{POP}} = 0.79$)	0.2148 ($\lambda_{\mathrm{POP}} = 0.99$)
Random	0.0071	0.0102	0.5002

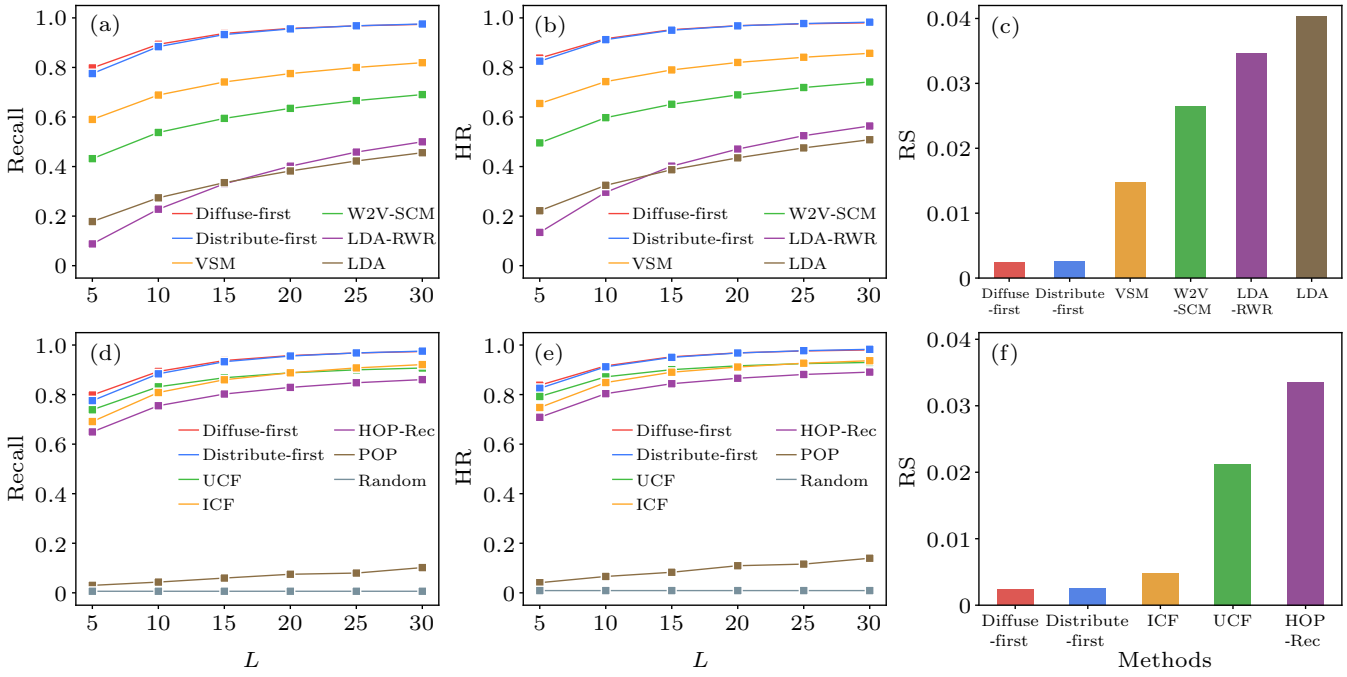


Fig. 5. Comparison of recommendation accuracy of the proposed methods with text-based methods (a)–(c) and non-text-based methods (d)–(f), under the corresponding optimal parameter combination in terms of recall (a) and (d), HR (b) and (e), and RS (c) and (f). Panels (a), (b), (d), and (e) show the recommendation performance under varying lengths of the recommendation list. In panel (f), the RS of the popularity-based and random recommendation is 0.2148 and 0.5002, respectively, which are omitted for better presentation.

5.2. Comparison with benchmark methods

We compare the performance of the two proposed methods with nine benchmark algorithms (including 4 text-based benchmarks and 5 non-text-based benchmarks) introduced in Subsection 4.2. Note that for methods that involve tunable parameters, the optimal parameters are determined by the average performance over 10 experiments, which assess the performance using the three aforementioned metrics. Figure 5

demonstrates the performance of the two proposed recommendation methods and nine benchmark methods under different lengths of recommendation lists and three evaluation metrics. Since the computation of the RS value is irrelevant to the recommendation list, we show the RS for each method with the optimal parameter combination in Figs. 5(c) and 5(f). Note that we omit the popularity-based and random recommendation in Fig. 5(f) in view of their performance being much worse than other methods.

We find that the distribute-first and diffuse-first recommendations perform comparably, with the latter showing a minor advantage. From Figs. 5(a)–5(c), it is evident that our proposed methods outperform all text-based methods across different evaluation metrics and recommendation list lengths. Within the four text-based methods themselves, VSM demonstrates the highest prediction accuracy, followed by W2V-SCM, whereas the two LDA-based methods exhibit the worst performance. This is because facing the sparsity of our corpus, TF-IDF techniques employed by VSM can mitigate the impact of low-frequency words by assigning low scores; yet both word2vec and LDA models require sufficient co-occurrence information of words to learn semantic meanings, making these methods less effective for rare words and not applicable for short texts such as abstracts.[64] Given the limited size of our corpus, we test the performance of W2V-SCM using pre-trained word and phrase embeddings that were trained on the part of the Google News dataset (around 100 billion words). This file can be accessed at code.google.com/archive/p/word2vec. The recall and hit metrics of W2V-SCM have experienced a rise of about 9%, which is not substantial enough to affect the comparison results. The advantage of W2V-SCM over LDA could be attributed to the generalization ability of the word2vec model on a domain-specific corpus, given that our corpus is solely composed of papers in the field of network science. This may be because, unlike LDA, which relies solely on word frequency, the word2vec model considers not just word frequency but also the contextual meaning and semantic relationships among terms. Regarding the two LDA-based methods, they exhibit different strengths depending on the size of the recommendation list: for smaller lists, standard LDA is more effective in identifying scholars with high semantic similarity; however, for larger lists, LDA-RWR outperforms by further incorporating relationships among scholars.

From Figs. 5(d)–5(f), the two multilayer network diffusion-based methods also show marked superiority over the non-text-based benchmark methods like UCF, ICF, and HOP-Rec. The results are consistent with consideration of all the three evaluation metrics and different lengths of the recommendation list. For instance, when $L = 20$, the highest recall values achieved by our proposed methods are over 7.62% higher than those of both CF approaches and 15.52% higher than that of HOP-Rec (see Table 2). Considering that the underperformance of these three methods could be due to their reliance on only one or two kinds of relations in the two-layer network, which limits their information utilization, we conduct a comparison under the same network structures. And the consistent results obtained further validate the advantage of our recommendation methods (see details in Appendix D).

When examining the specific limitations of each method, the inferiority of the CF approaches can be attributed to their limited coverage of potential reviewers. Specifically, UCF limits similar papers to those sharing connected scholars with the target paper, and then recommends scholars connected to these similar papers. ICF limits relevant scholars to those connected to the target paper, and then recommends scholars share connected papers with these relevant scholars. While in the proposed methods, resources diffuse among three subnetworks sequentially, scholars who have higher-order interactions with the target paper could also be recommended. On the other hand, although the HOP-Rec model captures higher-order interactions, it also introduces additional noise, resulting in inferior performance compared to UCF and ICF; another reason is that the random walk process in HOP-Rec is biased towards scholars with high degrees, which is inappropriate for the reviewer recommendation scenario. This also explains why the popularity-based recommendation is much worse than the above five methods. Thus, when choosing reviewers, the key factor is the relevance between the reviewers' research fields and the submitted manuscripts, because even the most experienced researchers cannot effectively review a paper that lies beyond their field of expertise. Besides, Fig. 5 and Table 2 reveal that the random recommendation achieves the lowest accuracy.

6. Conclusion and discussion

We put forward a multilayer network diffusion-based method to recommend reviewers for submitted papers. By using the relationship between scholars and their connected papers, we first build a scholar–paper bipartite network, based on which we consider the collaboration among scholars and the bibliographic coupling among papers, thus building a two-layer network. Based on the traditional mass diffusion algorithm on bipartite networks, we further consider the diffusion order and proportion on the two-layer network model. Through recommendation experiments on a real academic dataset which is extended from 70 reviewers in NetSci-X 2018 conference, we reveal that in general, the recommendation method which starts at the resource diffusion to papers performs better than the method that starts at the resource distribution among scholars. Besides, the nonzero optimal diffusion proportions indicate that introducing the collaboration

and bibliographic coupling relations helps to improve the recommendation accuracy. Most importantly, compared with the state-of-the-art methods that incorporate graph random walk and matrix factorization and methods that incorporate machine learning and natural language processing, the proposed methods outperform by more than 7.62% in recall, 5.66% in HR, and 47.53% in RS. The results prove that our methods can mine the reviewer–paper similarity more accurately. Our work provides a feasible and efficient way to recommend reviewers for papers in the peer review process without requiring textual information of the submitted manuscript and available reviewers. It can help to speed up the submission handling process and ease the burden on editors.

There are three main limitations of our work. Firstly, several practical constraints in the real peer review process, such as the conflicts of interest, are not taken into account in our proposed methods. One potential solution is incorporating a filtering mechanism for the recommended reviewers.

Secondly, we do not consider the cold-start problem in the main text, which refers to the insufficient information of a new coming manuscript. While the proposed method offers a partial solution by enabling new manuscripts to link to the network through bibliographic coupling relations with existing paper nodes, further exploration of underlying relationships holds great promise for enhancing and refining this solution.

Thirdly, due to the lack of real peer-review data, the performance evaluation of our method is currently limited to a small conference review dataset and an expanded dataset derived from it. However, we are actively overcoming this limitation by maintaining an online journal review platform and collecting more peer-review data for future research in recommendation systems. By leveraging large-scale real data, we anticipate that our proposed two-layer network model will effectively address the aforementioned limitations, making it a promising tool for reviewer recommendations.

Our work proves the effectiveness of the multilayer network diffusion-based methods on the reviewer recommendation problem, based on which studies can introduce more accessible information of papers and reviewers to improve the network structure for better recommendation accuracy. Beyond reviewer's recommendation, the research approach developed in this paper will find applications to information retrieval problems in more practical scenes.

Appendix A: Experiments on the NetSci-X 2018 review dataset

To test the performance of proposed methods in real peer review scenarios, we make experiments on a small review dataset, which records the reviewing procedures in NetSci-X 2018 conference involved 120 submissions and 61 reviewers. Considering the submissions for this conference are in the form of abstracts rather than full text, we further retrieve the published version of all manuscripts from the Scopus database to obtain the full lists of their references and authors.

As shown in Fig. A1, we first build the reviewer–manuscript two-layer network based on the proposed recommendation method. Similar as the scholar–paper two-layer network in Subsection 3.2, it is also composed of three different sub-networks: a co-reviewing network in the reviewer layer, a bibliographic coupling network in the manuscript layer, and a reviewing network between two layers. In the co-reviewing network, two reviewers are connected if they have commonly participated in the review of at least one manuscript. And in the bibliographic coupling network, there is a link between two submissions if they have at least one co-citing item, and the link weight represents the number of common references. For new coming manuscripts (like M4 in Fig. A1), the bibliographic coupling relations help them to connect with the reviewed ones (like M1–M3 in Fig. A1) in the network. In the reviewing network, we connect reviewers with submissions for which they have expressed a willingness to review. The willingness of reviewers is inferred from their historical reviewing and bidding records (in the bidding process of NetSci-X 2018 conference, for a given submission, a reviewer bidding "yes" means that the reviewer would like to review and "maybe" means that the reviewer can review. We consider these two options as reviewers' positive willingness). Table A1 shows the statistical properties of the three sub-networks.

Table A1. The statistical properties of the reviewing network, co-reviewing network, and bibliographic coupling network.

Networks	Nodes	Links	$\langle k \rangle$	
			$\langle k_r \rangle$	$\langle k_m \rangle$
Reviewing network	181	479	7.85	3.99
Co-reviewing network	61	585	19.18	–
Bibliographic coupling network	120	2,117	–	35.28

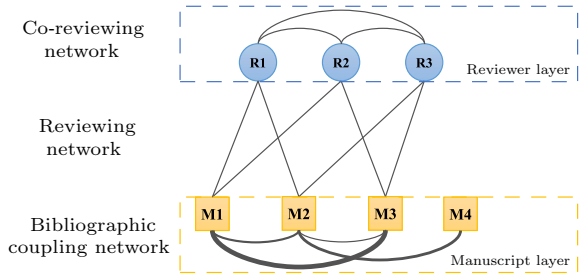
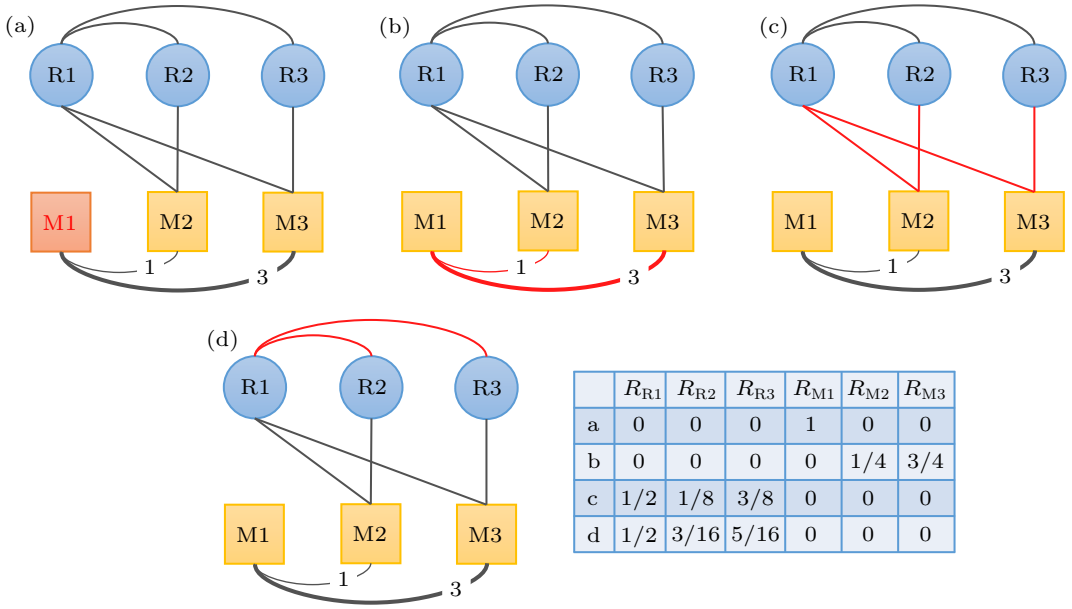


Fig. A1. Illustration of the reviewer–manuscript two-layer network. There are two layers composed of reviewer nodes and manuscript nodes, respectively, and three types of links, which represent reviewing, co-reviewing and bibliographic coupling relations, respectively. In the bibliographic coupling network, the thickness of each link is proportional to its weight, which is determined by the number of common references between two manuscripts. Links in the reviewing network and co-reviewing network are unweighted.



	R_{R1}	R_{R2}	R_{R3}	R_{M1}	R_{M2}	R_{M3}
a	0	0	0	1	0	0
b	0	0	0	0	1/4	3/4
c	1/2	1/8	3/8	0	0	0
d	1/2	3/16	5/16	0	0	0

Fig. A2. A simple example of the diffusion process on the reviewer–manuscript two-layer network. Manuscript M1 is the target manuscript. $p_2 = 1/2$. In each step, links in red are those involved in the diffusion process. The table records the resource value of each node after each step.

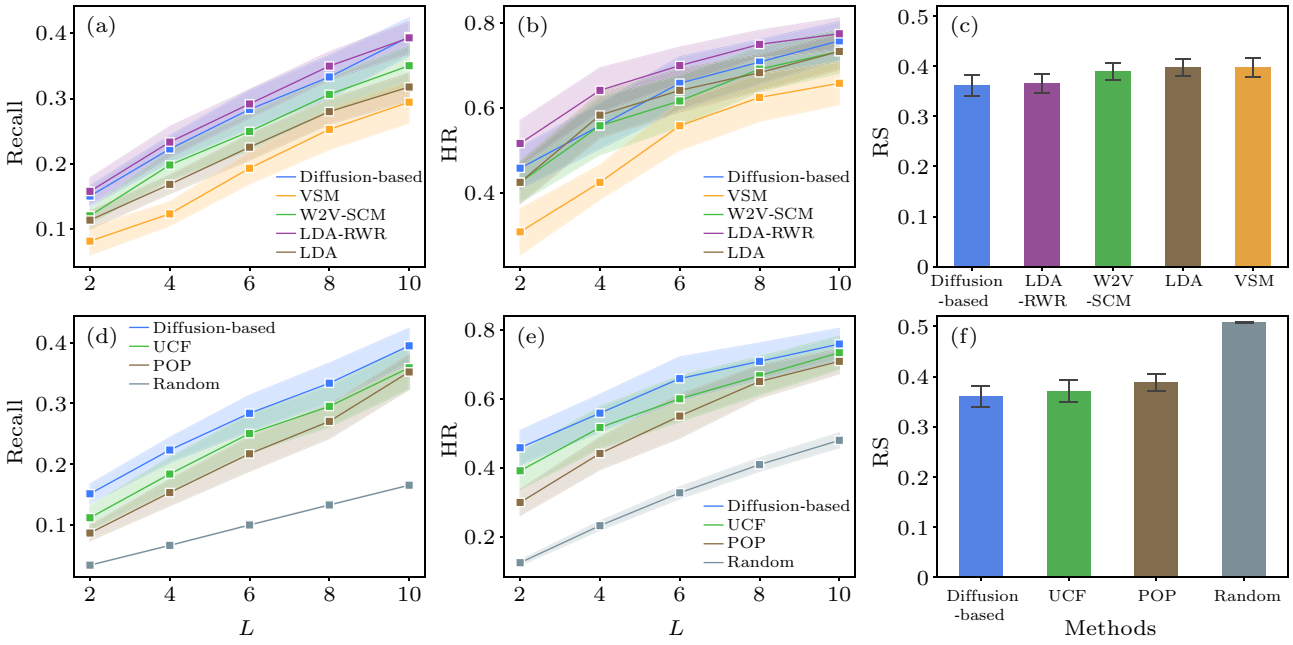


Fig. A3. Comparison of the recommendation performance of the proposed method with four text-based methods (a)–(c) and three non-text-based methods (d)–(f), in terms of recall (a) and (d), HR (b) and (e), and RS (c) and (f), on the NetSci-X 2018 dataset. The error bars indicate the standard error of the average performance.

When applying the multilayer network diffusion-based methods to the reviewer–manuscript two-layer network, in contrast to the scholar–paper two-layer network, the main challenge is that the new coming manuscript is not connected with any reviewer, which arises the cold-start problem. To make our methods executable for these papers, we adjust the diffusion process by assigning one unit of initial resources to the target manuscript. Then, the resource diffusion proceeds in the bibliographic coupling network, reviewing network and co-reviewing network successively. After that, we rank reviewers in descending order based on their resources, the top L reviewers form the recommendation list, which will be recommended as potential reviewers for the target manuscript. Figure A2 illustrates a simple example of the recommendation process for the target manuscript M1.

Our dataset is relatively small, so we choose 10-fold cross-validation to make experiments, where all 120 manuscripts are randomly divided into 10 sets with equal size. The experiments are carried out for 10 times separately, in which each set is used once for testing and the remaining nine sets are used for training. To simulate real peer-review scenarios, we remove all reviewing relationships of manuscripts in

the testing set, and then implement our method to recommend potential reviewers. All text-based methods are trained from the corpus described in Subsection 4.2.1, the corresponding results are shown in Figs. A3(a)–A3(c). ICF and HOP-Rec are excluded due to the cold-start problem, Figs. A3(d)–A3(f) only involve the comparisons with UCF, the popularity-based and random recommendation. Figure A3 shows the average results of the optimal performance on the 10 training/testing set splittings, which indicates that the proposed method has an evident advantage over non-text-based methods, and is only inferior to LDA-RWR among all text-based methods. The superiority of LDA-RWR comes from the detailed corpus, and the sparse reviewing network makes the proposed method slightly underperform. But the lowest RS still confirms the effectiveness of multilayer network diffusion-based methods in real peer review scenarios.

Appendix B: Derivation of Eqs. (1) and (2) in Subsection 4.1

Equations (1) and (2) in Subsection 4.1 reveal the relations between the resources before and after the diffusion process. Here we explain the derivation process. We denote the initial resources assigned to scholar α as R_α, and the final resources scholar γ received as R'_γ, other notations have been introduced in Subsection 3.2.

Derivation of Eq. (1)

Step (A): initial resources assignment. For scholar α, the initial resource is R_α.

Step (B): the resource redistribution among scholars. Let β be any scholar in the constructed network, β keeps $(1-p_2)$ proportion of the scholar's initial resources (R_β) and transfers the remaining p_2 proportion of resources to the scholar's coauthors. Meanwhile, β obtains resources from other scholars in the co-authorship network, so the resources of scholar β after step (B) are given by

$$R_\beta^{(\mathrm{B})} = (1-p_2)R_\beta + p_2 \sum_{\alpha=1}^{m} \frac{C_{\alpha\beta}R_\alpha}{k_\alpha^C}. \tag{B1}$$

Step (C): the diffusion from scholars to papers. Let i be any paper in the network, it receives resources from the connected scholars in the publishing network, that is

$$R_i^{(\mathrm{C})} = \sum_{\beta=1}^{m} \frac{A_{i\beta}R_\beta^{(\mathrm{B})}}{k_\beta^A}. \tag{B2}$$

Substitute Eq. (B1) into Eq. (B2), then we get

$$R_i^{(\mathrm{C})} = (1-p_2)\sum_{\beta=1}^{m} \frac{A_{i\beta}R_\beta}{k_\beta^A} + p_2 \sum_{\beta=1}^{m}\sum_{\alpha=1}^{m} \frac{A_{i\beta}C_{\alpha\beta}R_\alpha}{k_\beta^A k_\alpha^C}. \tag{B3}$$

Step (D): the resource redistribution among papers. Let j be any paper, it reserves $(1-p_1)$ proportion of its resources, and transfers the remaining p_1 proportion of resources to the connected papers in the bibliographic coupling network, proportionally based on the link weights. At the same time, it obtains resources from other papers, the total resources of j can be expressed as

$$R_j^{(\mathrm{D})} = (1-p_1)R_j^{(\mathrm{C})} + p_1 \sum_{i=1}^{n} \frac{B_{ij}R_i^{(\mathrm{C})}}{w_i^B}. \tag{B4}$$

Substitute Eq. (B3) into Eq. (B4), then we get

$$\begin{aligned} R_j^{(\mathrm{D})} =\ & (1-p_1)(1-p_2)\sum_{\beta=1}^{m} \frac{A_{j\beta}R_\beta}{k_\beta^A} \\ & + (1-p_1)p_2 \sum_{\beta=1}^{m}\sum_{\alpha=1}^{m} \frac{A_{j\beta}C_{\alpha\beta}R_\alpha}{k_\beta^A k_\alpha^C} \\ & + p_1(1-p_2)\sum_{i=1}^{n}\sum_{\beta=1}^{m} \frac{B_{ij}A_{i\beta}R_\beta}{w_i^B k_\beta^A} \\ & + p_1 p_2 \sum_{i=1}^{n}\sum_{\beta=1}^{m}\sum_{\alpha=1}^{m} \frac{B_{ij}A_{i\beta}C_{\alpha\beta}R_\alpha}{w_i^\beta k_\beta^A k_\alpha^C}. \end{aligned} \tag{B5}$$

Step (E): the diffusion from papers to scholars. For scholar γ, the scholar's resources are transferred from the connected papers in the publishing network, which are given by

$$R'_\gamma = \sum_{j=1}^{n} \frac{A_{j\gamma}R_j^{(\mathrm{D})}}{k_j^A}. \tag{B6}$$

We can employ Eq. (B5) to replace the term $R_j^{(\mathrm{D})}$ in Eq. (B6), and yield

$$\begin{aligned} R'_\gamma =\ & (1-p_1)(1-p_2)\sum_{j=1}^{n}\sum_{\beta=1}^{m} \frac{A_{j\gamma}A_{j\beta}R_\beta}{k_j^A k_\beta^A} \\ & + (1-p_1)p_2 \sum_{j=1}^{n}\sum_{\beta=1}^{m}\sum_{\alpha=1}^{m} \frac{A_{j\gamma}A_{j\beta}C_{\alpha\beta}R_\alpha}{k_j^A k_\beta^A k_\alpha^C} \\ & + p_1(1-p_2)\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\beta=1}^{m} \frac{A_{j\gamma}B_{ij}A_{i\beta}R_\beta}{k_j^A w_i^B k_\beta^A} \\ & + p_1 p_2 \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\beta=1}^{m}\sum_{\alpha=1}^{m} \frac{A_{j\gamma}B_{ij}A_{i\beta}C_{\alpha\beta}R_\alpha}{k_j^A w_i^\beta k_\beta^A k_\alpha^C}. \end{aligned} \tag{B7}$$

Finally, for Eq. (B7), we can further replace j in the first two terms on the right hand side with i, and replace β with α in the first and third terms on the right hand side, then Eq. (B7) becomes Eq. (1).

Derivation of Eq. (2)

Step (A): initial resources assignment. For scholar α, the initial resource is R_α.

Step (B): the diffusion from scholars to papers. Let i be any paper, it receives resources from the connected scholars in the publishing network, which can be represented as

$$R_i^{(\mathrm{B})} = \sum_{\alpha=1}^{m} \frac{A_{i\alpha}R_\alpha}{k_\alpha^A}. \tag{B8}$$

Step (C): the resource redistribution among papers. Let j be any paper, it reserves $(1-p_1)$ proportion of its resources, and transfers the remaining p_1 proportion of resources to the connected papers in the bibliographic coupling network, proportionally based on the link weights. Simultaneously, paper j obtains resources from other papers, we write the resources of paper j after step (C) as

$$R_j^{(\mathrm{C})} = (1-p_1)R_j^{(\mathrm{B})} + p_1 \sum_{i=1}^{n} \frac{B_{ij}R_i^{(\mathrm{B})}}{w_i^B}. \tag{B9}$$

Substitute Eq. (B8) into Eq. (B9), and we have

$$R_j^{(\mathrm{C})} = (1-p_1) \sum_{\alpha=1}^{m} \frac{A_{j\alpha}R_\alpha}{k_\alpha^A} + p_1 \sum_{i=1}^{n}\sum_{\alpha=1}^{m} \frac{B_{ij}A_{i\alpha}R_\alpha}{w_i^B k_\alpha^A}. \tag{B10}$$

Step (D): the diffusion from papers to scholars. Let β be any scholar, β obtains resources from the connected papers in the publishing network, which are given by

$$R_\beta^{(\mathrm{D})} = \sum_{j=1}^{n} \frac{A_{j\beta}R_j^{(\mathrm{C})}}{k_j^A}. \tag{B11}$$

Substitute Eq. (B10) into Eq. (B11), and we obtain

$$\begin{aligned} R_\beta^{(\mathrm{D})} = {} & (1-p_1) \sum_{j=1}^{n}\sum_{\alpha=1}^{m} \frac{A_{j\beta}A_{j\alpha}R_\alpha}{k_j^A k_\alpha^A} \\ & + p_1 \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\alpha=1}^{m} \frac{A_{j\beta}B_{ij}A_{i\alpha}R_\alpha}{k_j^A w_i^B k_\alpha^A}. \end{aligned} \tag{B12}$$

Step (E): the resource redistribution among scholars. For scholar γ, $(1-p_2)$ proportion of the scholar's resources is reserved, and the remaining p_2 proportion of resources is transferred to the scholar's coauthors. Simultaneously, γ obtains resources from other scholars in the co-authorship network, so the resources of scholar γ can be expressed as

$$R'_\gamma = (1-p_2)R_\gamma^{(\mathrm{D})} + p_2 \sum_{\beta=1}^{m} \frac{C_{\beta\gamma}R_\beta^{(\mathrm{D})}}{k_\beta^C}. \tag{B13}$$

According to Eq. (B12), we can replace the terms $R_\gamma^{(\mathrm{D})}$ and $R_\beta^{(\mathrm{D})}$ in Eq. (B13). This yields

$$\begin{aligned} R'_\gamma = {} & (1-p_2)(1-p_1) \sum_{j=1}^{n}\sum_{\alpha=1}^{m} \frac{A_{j\gamma}A_{j\alpha}}{k_j^A k_\alpha^A} R_\alpha \\ & + (1-p_2)p_1 \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\alpha=1}^{m} \frac{A_{j\gamma}B_{ij}A_{i\alpha}}{k_j^A w_i^B k_\alpha^A} R_\alpha \\ & + p_2(1-p_1) \sum_{\beta=1}^{m}\sum_{j=1}^{n}\sum_{\alpha=1}^{m} \frac{C_{\beta\gamma}A_{j\beta}A_{j\alpha}}{k_\beta^C k_j^A k_\alpha^A} R_\alpha \\ & + p_2 p_1 \sum_{\beta=1}^{m}\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\alpha=1}^{m} \frac{C_{\beta\gamma}A_{j\beta}B_{ij}A_{i\alpha}}{k_\beta^C k_j^A w_i^B k_\alpha^A} R_\alpha. \end{aligned} \tag{B14}$$

Finally, for Eq. (B14), we can further replace j in the first and third terms on the right hand side with i, then Eq. (B14) becomes Eq. (2).

Appendix C: Evaluation on different training/testing set splittings

To further validate the stability of experimental results, we repeat the experiments introduced in Subsection 5.2, with two different training/testing set splittings — 95%/5% and 85%/15%. The results for the 95%/5% splitting are shown in Fig. C1 and for the 85%/15% splitting, see Fig. C2. The results are averaged over 10 experiments, both Figs. C1 and C2 are consistent with Fig. 5.

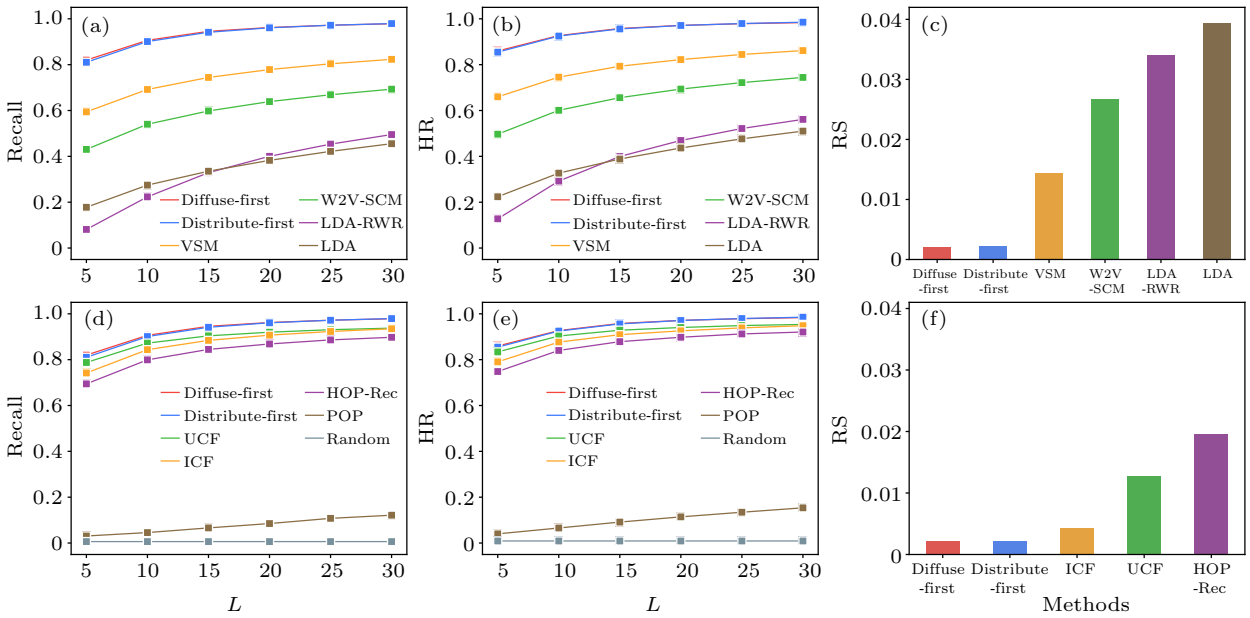


Fig. C1. Comparison of the average performance of the studied recommendation methods in terms of recall (a) and (d), HR (b) and (e), and RS (c) and (f), with the 95%/5% splitting of the training and testing set. Panels (a), (b), (d) and (e) show the recommendation performance under varying lengths of the recommendation list. In panel (f), the RS of the popularity-based and random recommendation is 0.1944 and 0.5000, respectively, which are omitted for better presentation. The relative improvement of the proposed methods over benchmark methods is similar as in Fig. 5.

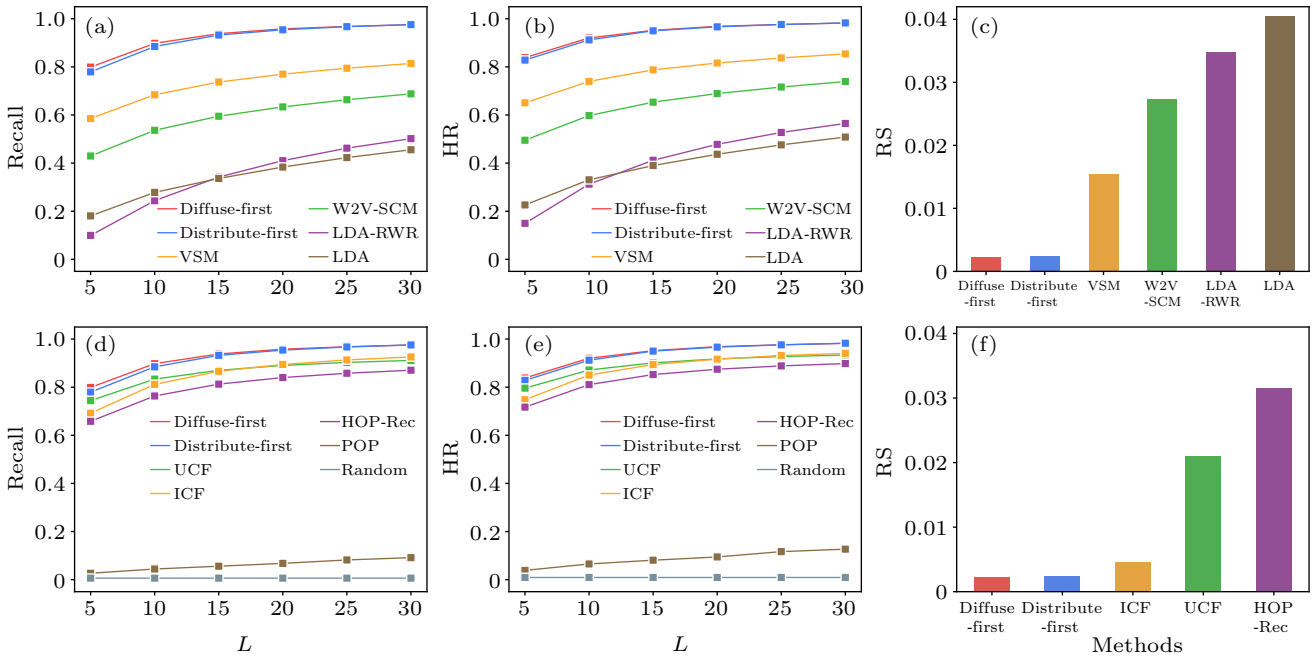


Fig. C2. Comparison of the average performance of the studied recommendation methods in terms of recall (a) and (d), HR (b) and (e), and RS (c) and (f), with the 85%/15% splitting of the training and testing set. Panels (a), (b), (d), (e) show the recommendation performance under varying lengths of the recommendation list. In panel (f), the RS of the popularity-based and random recommendation is 0.2163 and 0.5004, respectively, which are omitted for better presentation. The relative improvement of the proposed methods over benchmark methods is similar as in Fig. 5.

Appendix D: Recommendation performance of non-text-based methods under the same network structures

According to the description of benchmark methods in Subsection 4.2, we can find that none of UCF, ICF and HOP-Rec exploit the complete structures of the constructed two-layer network. In particular, UCF depends on the publishing and bibliographic coupling relations, ICF is based on the publishing and co-authorship relations, and HOP-Rec only utilizes the publishing relations.

To demonstrate that the advantages of proposed methods do not solely stem from utilizing more information, we further compare the performance of the proposed methods with UCF, ICF and HOP-Rec under the same network structures, which is achieved by adjusting the diffusion proportions, p_1 and p_2 in our methods. Specifically, we set $p_2 = 0$ when comparing with

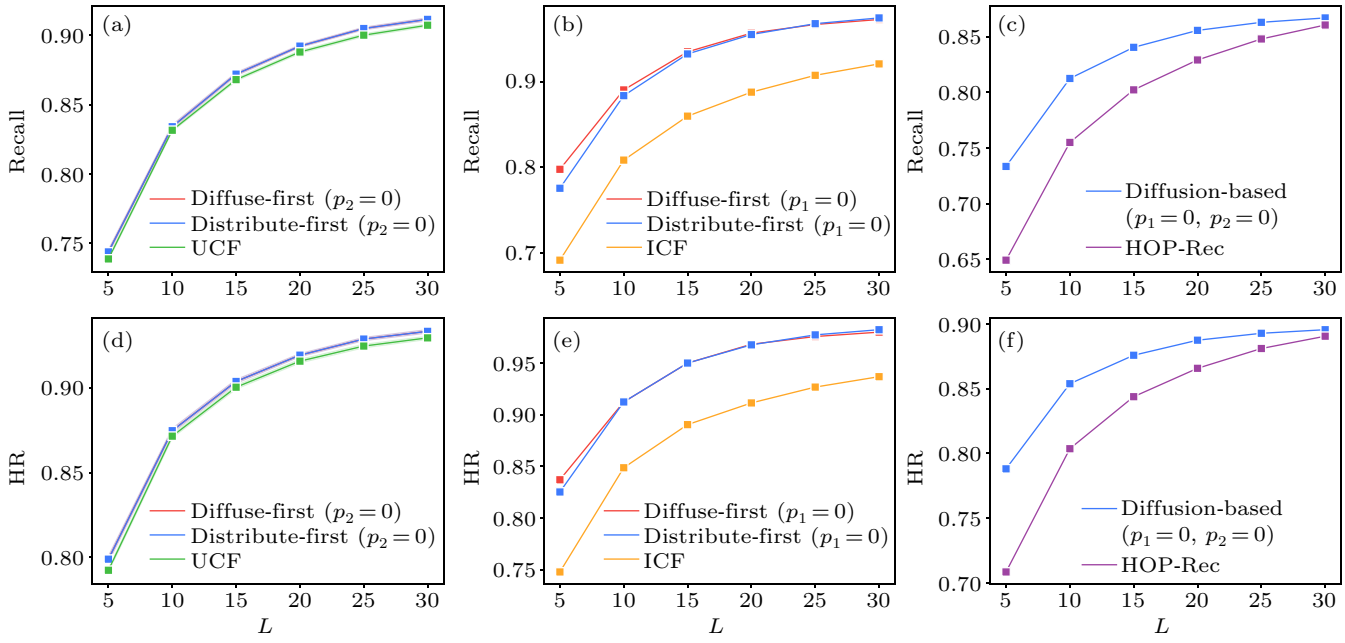


Fig. D1. Comparison of the average performance of the proposed methods and the three non-text-based methods under the same network structures, in terms of recall (a)–(c) and HR (d)–(f). When $p_2 = 0$, the diffuse-first and distribute-first methods achieve similar performance, resulting in overlapping lines in panels (a) and (d).

UCF, $p_1 = 0$ when comparing with ICF, $p_1 = 0$ and $p_2 = 0$ when comparing with HOP-Rec. The results are averaged over 10 experiments with the 90%/10% training/testing splitting, as shown in Fig. D1. Under varying lengths of the recommendation list, the average performance of proposed methods is around 0.47% higher than that of the UCF in terms of recall and HR. Compared with the ICF, the advantage is larger, our methods perform around 8.35% higher in recall and around 6.83% higher in HR. As for the HOP-Rec, our methods outperform by around 5.17% in recall and 4.29% in HR. These reveal that the advantages of the proposed methods over the three benchmark methods still hold under the same network structures.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. T2293771) and the New Cornerstone Science Foundation through the XPLORER PRIZE.

References

[1] Patil A H and Mahalle P N 2020 *Procedia Comput. Sci.* **171** 709
[2] Zhao X Q and Zhang Y S 2022 *Inf. Process. Manag.* **59** 103028
[3] Zaharie M A and Seeber M 2018 *Scientometrics* **117** 1587
[4] Duan Z, Tan S C, Zhao S, Wang Q Q, Chen J and Zhang Y P 2019 *Neurocomputing* **366** 97
[5] Liu X, Suel T and Memon N 2014 *Proceedings of the 8th ACM Conference on Recommender Systems*, October 6–10, 2014, Foster City, USA, p. 25
[6] Di Mauro N, Basile T M and Ferilli S 2005 *18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, June 22–24, 2005, Bari, Italy, p. 789
[7] Tan S C, Duan Z, Zhao S, Chen J and Zhang Y P 2021 *Inform. Retrieval J.* **24** 175
[8] Yang J H, Chen C M, Wang C J and Tsai M F 2018 *Proceedings of the 12th ACM Conference on Recommender Systems*, October 2–7, 2018, Vancouver, Canada, p. 140
[9] Dumais S T and Nielsen J 1992 *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, June 21–24, 1992, Copenhagen, Denmark, p. 233
[10] Tayal D K, Saxena P C, Sharma A, Khanna G and Gupta S 2014 *Appl. Intell.* **40** 54
[11] Kou N M, Leong Hou U, Mamoulis N and Gong Z 2015 *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, May 31–June 4, 2015, Melbourne, Australia, p. 2031
[12] Zhao S, Zhang D, Duan Z, Chen J, Zhang Y P and Tang J 2018 *Scientometrics* **115** 1293
[13] Yarowsky D and Florian R 1999 *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, June 21–22, 1999, Maryland, USA, p. 220
[14] Mirzaei M, Sander J and Stroulia E 2019 *Inf. Process. Manag.* **56** 858
[15] Yang C, Liu T T, Yi W J, Chen X H and Niu B 2020 *Appl. Soft Comput.* **94** 106483
[16] Zhang D, Zhao S, Duan Z, Chen J, Zhang Y P and Tang J 2020 *ACM Trans. Inf. Syst.* **38** 5
[17] Kalmukov Y 2020 *Proceedings of the 21st International Conference on Computer Systems and Technologies*, June 19–20, 2020, Ruse, Bulgaria, p. 229
[18] Abduljaleel A Q, Naser M A and Al-mamory S O 2021 *Turk. J. Comput. Math. Educ.* **12** 619
[19] Kalmukov Y 2012 *Comput. Sci. Inf. Syst.* **9** 763
[20] Protasiewicz J, Pedrycz W, Kozłowski M, Dadas S, Stanisławek T, Kopacz A and Gałężewska M 2016 *Knowl. Based Syst.* **106** 164
[21] Blei D M, Ng A Y and Jordan M I 2003 *J. Mach. Learn. Res.* **3** 993
[22] Mimno D and McCallum A 2007 *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 12–15, 2007, San Jose, USA, p. 500
[23] Rosen-Zvi M, Griffiths T L, Steyvers M and Smyth P 2004 *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, July 7–11, 2004, Banff, Canada, p. 487
[24] Jin J, Geng Q, Mou H K and Chen C 2019 *J. Inf. Sci.* **45** 554
[25] Peng H W, Hu H J, Wang K Q and Wang X L 2017 *International Conference on Database Systems for Advanced Applications*, March 27–30, 2017, Suzhou, China, p. 145
[26] Ogunleye O, Ifebanjo T, Abiodun T and Adebiyi A 2017 *Covenant University Conference on E-Governance in Nigeria*, 2017, Ota, Nigeria, p. 211
[27] Yin H Z, Cui B, Lu H and Zhao L 2016 2016 *Conference on Technologies and Applications of Artificial Intelligence*, November 25–27, 2016, Hsinchu, China, p. 1
[28] Jin J, Niu B Z, Ji P and Geng Q 2020 *Ann. Oper. Res.* **291** 409
[29] Nie L, Davison B D and Qi X G 2006 *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 6–11, 2006, Seattle, USA, p. 91
[30] Rodriguez M A and Bollen J 2008 *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, October 26–30, 2008, Napa Valley, USA, p. 319
[31] Goldsmith J and Sloan R H 2007 *Proceedings of AAAI Workshop on Preference Handling for Artificial Intelligence*, July 22–23, 2007, Vancouver, Canada, p. 53
[32] Tong H H, Faloutsos C and Pan J Y 2006 *Proceedings of the Sixth International Conference on Data Mining*, December 18–22, 2006, Hong Kong, China, p. 613
[33] Xu Y H and Du Y W 2013 2013 *Sixth International Conference on Business Intelligence and Financial Engineering*, November 14–16, 2013, Hangzhou, China, p. 552
[34] Pradhan T, Sahoo S, Singh U and Pal S 2021 *Expert Syst. Appl.* **169** 114331
[35] Tang X J and Zhang Z W 2008 2008 *IEEE International Conference on Systems, Man and Cybernetics*, October 12–15, 2008, Suntec, Singapore, p. 102
[36] Liu H, Wang B J, Lu J A and Li Z Y 2021 *Acta Phys. Sin.* **70** 056401 (in Chinese)
[37] Yang F, Zhu J, Lun J Q, Zheng Z T, Tang Y and Wu J 2018 2018 *IEEE 22nd International Conference on Computer Supported Cooperative Work in Design*, May 9–11, 2018, Nanjing, China, p. 247
[38] Liu H, Shang Z C, Ren Z Y, Li Y, Zeng Z G and Lu J A 2022 *Sci. China Technol. Sci.* **65** 1493
[39] Su X Y and Khoshgoftaar T M 2009 *Adv. Artif. Intell.* **2009** 1
[40] Lü L Y, Medo M, Yeung C H, Zhang Y C, Zhang Z K and Zhou T 2012 *Phys. Rep.* **519** 1
[41] Rigaux P 2004 *Proceedings of the 2004 ACM Symposium on Applied Computing*, March 14–17, 2004, Nicosia, Cyprus, p. 1682
[42] Li X L and Watanabe T 2013 *Procedia Comput. Sci.* **22** 633
[43] Conry D, Koren Y and Ramakrishnan N 2009 *Proceedings of the Third ACM Conference on Recommender Systems*, October 23–25, 2009, New York, USA, p. 357
[44] Rose M E and Kitchin J R 2019 *SoftwareX* **10** 100263
[45] Liu J G, Zhou Q, Guo Q, Yang Z H, Xie F and Han J T 2017 *Sci. Rep.* **7** 10755
[46] Fu G Y, Chen F E, Liu J G and Han J T 2019 *Physica A* **525** 409
[47] Zhou T, Ren J, Medo M and Zhang Y C 2007 *Phys. Rev. E* **76** 046115
[48] Zhou T, Kuscsik Z, Liu J G, Medo M, Wakeling J R and Zhang Y C 2010 *Proc. Natl. Acad. Sci. USA* **107** 4511
[49] Liu J G, Zhou T and Guo Q 2011 *Phys. Rev. E* **84** 037101
[50] Guo Q, Leng R, Shi K and Liu J G 2012 *Eur. Phys. J. B* **85** 1
[51] Guo Q, Song W J, Hou L, Zhang Y L and Liu J G 2014 *Physica A* **401** 15
[52] Liu J G, Wang B H and Guo Q 2009 *Int. J. Mod. Phys. C* **20** 285
[53] Pan X, Deng G S and Liu J G 2010 *Phys. Procedia* **3** 1867
[54] Nie D C, An Y H, Dong Q, Fu Y and Zhou T 2015 *Physica A* **421** 44
[55] Deng X F, Zhong Y S, Lü L Y, Xiong N X and Yeung C H 2017 *Inf. Sci.* **417** 420
[56] Karimzadehgan M and Zhai C X 2012 *Inf. Process. Manag.* **48** 725
[57] Röder M, Both A and Hinneburg A 2015 *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, January 31–Febuary 6, 2015, Shanghai, China, p. 399

[58] Sidorov G, Gelbukh A, Gómez-Adorno H and Pinto D 2014 *Comput. Sist.* **18** 491
[59] Sarwar B, Karypis G, Konstan J and Riedl J 2001 *Proceedings of the 10th International Conference on World Wide Web*, May 1–5, 2001, Hong Kong, China, p. 285
[60] Binesh N and Rezghi M 2014 2014 *6th Conference on Information and Knowledge Technology*, May 28–30, 2014, Tehran, Iran, p. 146
[61] Aiolli F 2013 *Proceedings of the 7th ACM Conference on Recommender Systems*, October 12–16, 2013, Hong Kong, China, p. 273
[62] Chen C M, Wang T H, Wang C J and Tsai M F 2019 *Proceedings of the 13th ACM Conference on Recommender Systems*, September 16–20, 2019, Copenhagen, Denmark, p. 582
[63] Zhao Z Y, Sheng Y Q, Zhu M and Wang J L 2018 *IEEE Access* **6** 67070
[64] Anjum O, Gong H Y, Bhat S, Hwu W M and Xiong J J 2019 *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, November 3–7, 2019, Hong Kong, China, p. 518