IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 12, 2018, accepted July 17, 2018, date of publication August 13, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2865115

Low-Rank and Sparse Matrix Factorization for Scientific Paper Recommendation in Heterogeneous Network

TAO DAI[1], TIANYU GAO[1], LI ZHU[1], XIAOYAN CAI[2], AND SHIRUI PAN[3]

[1]School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2]School of Automation, Northwestern Polytechnical University, Xi'an 710072, China
[3]Centre for Artificial Intelligence, University of Technology Sydney, Sydney, NSW 2007, Australia

Corresponding author: Li Zhu (zhuli@xjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Project 61373046 and in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Project S2015YFJM2129.

ABSTRACT With the rapid growth of scientific publications, it is hard for researchers to acquire appropriate papers that meet their expectations. Recommendation system for scientific articles is an essential technology to overcome this problem. In this paper, we propose a novel low-rank and sparse matrix factorization-based paper recommendation (LSMFPRec) method for authors. The proposed method seamlessly combines low-rank and sparse matrix factorization method with fine-grained paper and author affinity matrixes that are extracted from heterogeneous scientific network. Thus, it can effectively alleviate the sparsity and cold start problems that exist in traditional matrix factorization based collaborative filtering methods. Moreover, LSMFPRec can significantly reduce the error propagated from intermediate outputs. In addition, the proposed method essentially captures the low-rank and sparse characteristics that exist in scientific rating activities; therefore, it can generate more reasonable predicted ratings for influential and uninfluential papers. The effectiveness of the proposed LSMFPRec is demonstrated by the recommendation evaluation conducted on the AAN and CiteULike data sets.

INDEX TERMS Paper recommendation, low rank and sparse matrix factorization, heterogeneous network.

I. INTRODUCTION

With the rapid development of information science, people are currently suffering from information overload problem. Recommendation systems, which allow users to find what they want and enable platforms to provide users what they might like, can significantly alleviate this problem. Due to their necessity and efficiency, recommendation systems have been widely applied and achieved success in many fields such as e-commerce [1], multimedia [2], social networks [3], [4] and web services [5].

Scientific paper recommendation is a particular recommendation service provided for researchers. When doing research, people are usually overwhelmed by the large amount of scholarly literature, which leads to a laborious and time-consuming search task for papers. Therefore, a timely and effective scientific paper recommendation system can significantly improve the work efficiency of researchers. According to different usages, paper recommendation can be divided into two types: personalized recommendation [6]–[9], [16]–[20], [50] and passive recommendation [10]–[15]. A personalized recommendation system requires user initiative by providing some text information, e.g., an entire manuscript or part of it, and it returns an article list that is related to the provided text. Typical examples of this type are academic search engines [11] and citation recommendation systems [7]–[9], [16], [17], [19]. However, a small amount of text may be too short or too ambiguous. Moreover, asking an author to provide a manuscript is sometimes impractical. Unlike personalized recommendation, passive recommendation does not require user involvement. It recommends articles according to the users' historical activities. For example, academic social networking sites, such as ResearchGate[1] and Academia,[2] recommend articles according to the historical publications of a researcher. Therefore, the articles recommended by passive recommendation

[1]www.researchgate.net
[2]www.academia.edu

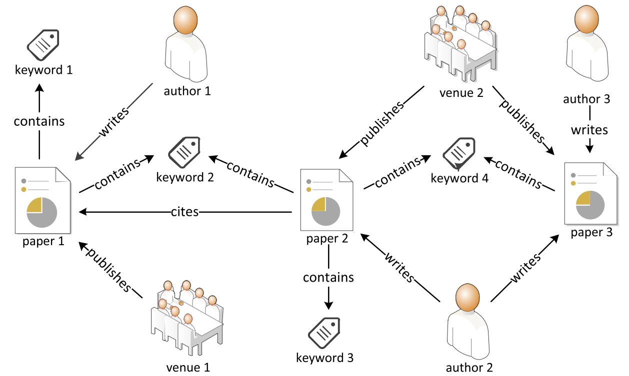


FIGURE 1. **An example of a heterogeneous bibliographic network.**

can provide researchers with a panoramic view of knowledge that matches their background, and it benefits long-term studies for researchers. In this paper, we focus on passive recommendation.

Collaborative filtering (CF), which automatically predicts the interests of a specific user based on the collective historical rating records of similar users or items, has been extensively studied in the field of paper recommendation [11], [21], [22], [28]. The most representative approach of CF is matrix completion [21], [49]. This approach decomposes the original rating matrix into two low-rank matrixes with a joint latent factor space. One matrix represents the latent interests of users, and the other represents the possessed factors of items. The recommendation results are thus obtained by the inner products of user vectors and item vectors. In reality, this approach usually suffers from the sparsity and cold start problem because the number of interactions between users and items is usually limited. A suitable solution for the above problem is adding more related information. In a scholarly dataset, there are various types of nodes and relations in addition to the author and paper. Therefore, the dataset is usually formed as a heterogeneous network. The bibliographic network shown in Fig. 1 is an example. There are four types of objects: paper, author, venue and keyword. These objects are also connected by various relationships, such as venue-publishes-paper relationship between venues and papers, and paper-contains-keyword relationship between papers and keywords. To utilize these various kinds of relationships, some variants of CF have been proposed by jointly decomposing other relationships [13]–[17], [20], [22]. However, the above methods have two drawbacks. First, the predicted rating is recovered from intermediate matrixes, which makes the error generated by intermediate values propagate to the final prediction. Second, these methods display a lack of interpretability due to the uninterpretable low dimensions.

In recent years, low-rank and sparse matrix factorization (LSMF) methods have gained increasing attention in many research fields [23]–[25]. Compared with traditional matrix decomposition in CF, some LSMF methods, such as GoDec [23] and RPCA [24], factorize the original matrix into a low-rank matrix and a sparse matrix. Matrix completion is then completed by adding the two matrixes. Unlike vector inner product, addition involves less computation. Furthermore, most values in the completed matrix are even directly obtained from the low-rank matrix because the sparse matrix is formed mostly by zero values. Therefore, LSMF can significantly reduce the error propagated by intermediate outputs. For scientific paper recommendation, LSMF also has more interpretability due to the natural low-rank and sparse characteristics of scientific article rating activities. Let us take citing as an example. Some influential articles are usually co-cited by authors who work in a similar research field. For example, nearly all authors who work on citation recommendation fields will cite [9], for this is the first paper that formally proposes the task definition of citation recommendation. Another example is that nearly all papers that are related to topic model cite LDA [26] because it is the most popular topic model. According to this similar citing pattern, the citing matrix for influential articles demonstrates low-rank character. Further, uninfluential articles are usually less cited, and the citing pattern for these papers shows more randomness. Therefore, the citing matrix for uninfluential papers shows sparse character. A more intuitive explanation is demonstrated in Fig. 2. We can see that by factoring the original rating matrix, influential articles p_1 and p_4 are captured by a low-rank matrix, while uninfluential articles p_2, p_3 and p_5 are captured by a sparse matrix. Traditional matrix decomposition in CF fails to capture these low-rank and sparse characteristics when recommending papers. Zhao *et al.* [27] proposed a low-rank and sparse matrix decomposition algorithm for movie and food recommendations. However, they directly decomposed a rating matrix into a single low-rank and sparse matrix, which can not reveal the characteristics of rating in scientific work. Moreover, their method cannot utilize various and valuable link information in heterogeneous scientific network, and also suffers from the sparsity and cold start problem.

In this paper, we propose a novel low-rank and sparse matrix factorization based paper recommendation (LSMF-PRec) method for authors. We extract fine-grained paper and author affinity matrixes from heterogeneous scientific network and seamlessly combine these useful relations into the learning process of LSMF. The proposed method can not only utilize abundant link information in the heterogeneous scientific network but also remedy the sparsity and cold start problems that existed in traditional CF. LSMFPRec can also significantly prevent the error generated by intermediate outputs. In addition, the rating characteristics of scientific articles can be directly revealed from the decomposed low-rank and sparse matrixes, which is beneficial for generating more suitable predicted ratings. LSMFPRec can also apply bilateral random projections (BRP) [39], which significantly accelerates the computation speed of matrix factorization.

The main contributions of this paper are summarized as follows.

(1) We propose a novel paper recommendation method with low-rank and sparse matrix factorization. As far as

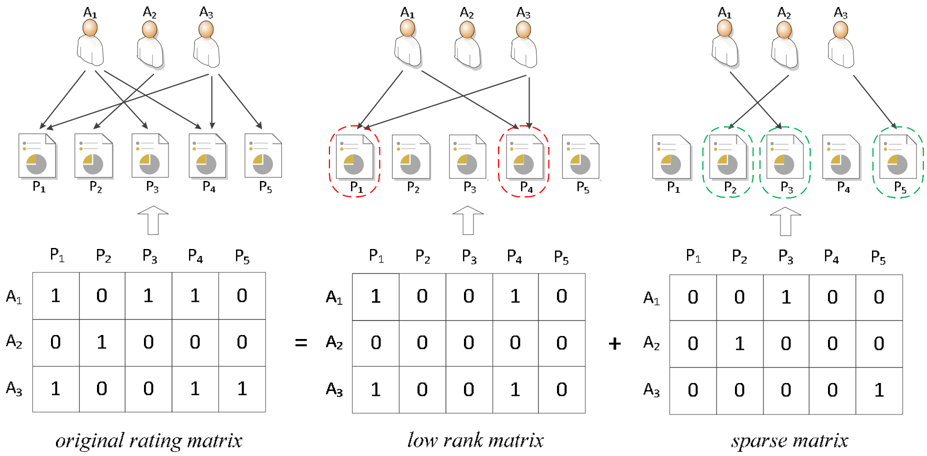


FIGURE 2. An example of low-rank and sparse characteristics in author rating activity.

we know, this is the first work that recommends papers by applying the LSMF method.

(2) We extend the original LSMF with link information in a heterogeneous scientific network, and prove the correctness and convergence of the learning process.

(3) Thorough experimental studies on the AAN and CiteULike datasets are performed to validate the effectiveness of the proposed method.

The reminder of the paper is organized as follows. Section II reviews related work. Section III presents the construction of paper and author affinity matrixes. Section IV provides a detailed description of the proposed method. Section V presents the experimental results and analysis. The paper is concluded in Section VI.

II. RELATED WORK

A. PAPER RECOMMENDATION

In 1998, Bollacker *et al.* [10] introduced the first scientific paper recommender system which is part of the CiteSeer project. Since then, many different methods have been employed for paper recommendation in the literature. Some systems also recommended "citations". However, in our opinion, differences between papers and citations are marginal. In general, paper recommendation methods can be divided into four categories: CF, content-based filtering (CBF), graph-based approaches and hybrid approaches.

Pennock *et al.* [22] considered that a user's ratings of unseen items are affected by the rating frequency of other users, and proposed a paper recommendation method, called personality diagnosis (PD), by using a Bayesian network. PD can be considered as a particular user-based CF approach. McNee *et al.* [9] took the citations of an author as positive votes for a paper and applied CF to recommend scientific papers. Yang *et al.* [28] combined memory based CF and ranking to predict the preference of a user towards articles. Liu *et al.* [17] considered co-concurrency to be a vital factor and proposed a context-based collaborative filtering method for paper recommendation. However, the above CF approaches suffer from cold start and sparsity problems. To overcome these disadvantages, some researchers explored recommendation through CBF. Unlike CF approach that only utilizes rating relationships, CBF explores information of user and item from the text content; thus, it is less affected by the above problems of CF. Sugiyama and Kan [12] built author profiles from published papers lists and recommended scholarly papers by capturing author research preferences. The author profile is enhanced through past publications that cited the work of the author. Alzoghbi *et al.* [57] proposed a learning-to-rank based CBF (LRCBF) method for paper recommendation. They developed two different validation mechanisms to examine pair-wise preferences, and applied Rank SVM [60] to predict suitable papers for author. Wang and Blei [13] proposed CTR model to recommend articles by combining a topic model with collaborating filter. CTR utilizes LDA to estimate latent topics for articles and matrix factorization to infer user-item relations. Many works followed CTR by adding more relations [14]–[16]. Wang *et al.* [55] proposed a deep learning based CBF, named collaborative deep learning (CDL), to extend CTR method. CDL uses stacked denoising autoencoder (SDAE) [56] to learn the deep representation of items, and is able to perform collaborative filtering for the rating matrix simultaneously. Despite its success, CBF approaches are limited by problems in traditional information retrieval, such as semantic ambiguity. In addition, the estimation of profile of authors and articles is usually time-consuming. To alleviate the above problem, Sharma *et al.* [58] proposed a concept-based paper recommendation approach (ConceptPRec). The method uses Paragraph Vector [59] to learn deep representations of papers, then calculates the similarity between candidate papers and user interested papers to perform recommendation.

Graph model is another widely applied method in paper recommendation field. Gori and Pucci [18] constructed a homogenous citation graph from bibliographic dataset and applied PageRank algorithm to recommend scientific papers. Meng *et al.* [6] referred topic as a node type and applied a random walk algorithm to recommend scientific papers from a four-layer heterogeneous graph. Guo *et al.* [7] extracted a fine-grained co-authorship from a citation graph and recommended papers by a graph-based paper ranking in multi-layered graph. They further expanded the ranking approach with mutually reinforced learning for personalized citation recommendation [8]. However, the major drawback of graph-based approach is the high time complexity when applied to large graphs. Moreover, the topic shift problem in graph model will retrieve many irrelevant results [29]. The previous introduced approaches may be combined in hybrid approaches. Torres *et al.* [11] proposed a recommendation system, named TechLens, that explores both the social relationships and the content of paper. TechLens is consists of three CBF variations, two CF variations, and five hybrid approaches. Ren *et al.* [19] assumed that each author has their own citation pattern and proposed a hybrid citation recommendation method ClusCite. The method combines a cluster and graph propagation approach to learn relativity and importance for recommended papers. Lee *et al.* [20] proposed a hybrid paper recommendation system that combines a content-based approach and a graph-based approach. The recommended papers are obtained by the weighted results of two approaches. Recently, there are emerging graph embedding [51], [52] or mining algorithms [53], [54] for graph analytics, which can be adapted to paper recommendation as well.

B. LOW-RANK AND SPARSE MATRIX FACTORIZATION

In recent years, the low-rank and sparse matrix factorization (LSMF) problem has attracted considerable attention in many fields, including video surveillance [30], [31], low-rank textures [32], image processing [33]–[35] and computer vision [36], [37]. Halko *et al.* [38] proposed randomized approximate matrix decomposition and demonstrated that a matrix can be well compressed by random sampling on column space. This revelatory approach provides an approximation of SVD/PCA with fast speed. Candès *et al.* [24] proved that the low-rank and sparse parts of a matrix can be disentangled exactly by convex programming and proposed robust principal component analysis (RPCA). Compared with former approaches that only consider low-rank components or sparse components, RPCA provides a unique separation of low-rank data and sparse noises. However, RPCA cannot predefine the rank of low-rank matrix and the sparsity of noise. To address this issue, Zhou and Tao [23] proposed the GoDec algorithm for LSMF. In addition to controllable rank and sparsity, GoDec applies bilateral random projections (BRP) [39] to increase the convergence speed. The decomposition of GoDec usually converges within 10∼15 iterations. These characteristics make GoDec a good choice for recommendation systems.

In recommendation area, Ning and Karypis [40] introduced a sparsity coefficient into the original CF and proposed sparse linear method (SLIM) for product and movie recommendation. The sparsity in SLIM is controlled by ℓ_1-norm of item matrix, and the optimization problem is solved by coordinate descent and soft thresholding [41]. However, SLIM did not consider the low rank character in rating activity, and the results will be affected by the error propagated from intermediate outputs as in CF. Zhao *et al.* [27] proposed a low-rank and sparse matrix completion (LSMC) method to obtain a low-rank and sparse predicted rating for food and movie recommendation. Unlike SLIM, there is only one low-rank and sparse matrix learned in LSMC; thus, it can remedy the error introduced by intermediate outputs. However, the assumption of LSMC seems unsuitable for paper recommendation, and it is hard to utilize various link information in a bibliographic network. Moreover, due to the regulation by Lagrange multiplier, LSMC cannot control the value of sparsity accurately.

III. FINE-GRAINED AFFINITY MATRIX CONSTRUCTION FROM HETEROGENEOUS BIBLIOGRAPHIC NETWORK

In a given heterogeneous bibliographic network, the edge weights between vertexes are usually all binary. For example, if paper p_i cites paper p_j, the weight of edge between them is 1; otherwise, it is 0. Directly applying these binary values as a relevance measurement of vertexes is irrational. There are two reasons. First, a zero value does not mean that there is no relation between vertexes. Take paper citation relations as an example. The reason why paper p_i did not cite p_j might be that the author was not aware of it, rather than p_j is irrelevant to p_i. Second, some important latent correlations between vertexes cannot be directly revealed by links. In a heterogeneous bibliographic network, papers contain abundant text content in addition to links, and these texts contain their own contextual features, such as semantic and syntactic information. This information is crucial when considering the correlations of papers. It is also obvious that the research interests and communities should be taken into account when measuring the relations of authors rather than considering only binary coauthor relations. Therefore, extracting fine-grained relations of rich information nodes is critical and essential for heterogeneous bibliographic network. In this section, we extract a fine-grained paper affinity matrix and an author affinity matrix. These affinity matrixes will be integrated into the low-rank and sparse matrix factorization process in Section IV.

A. CONSTRUCTION OF FINE-GRAINED PAPER AFFINITY MATRIX

Given a heterogeneous bibliographic network, we can extract three types of relations between papers: immediate relation, mediate relation and latent relation.

1) IMMEDIATE RELATION BETWEEN PAPERS

In a bibliographic network, the immediate relation can be obtained directly from citing links between papers.

Define W_{pp}^c as the citing matrix, and each element of it is calculated as follows.

$$w_{p_ip_j}^c = \begin{cases} 1 & \text{if } p_i \text{ is cited by } p_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

2) MEDIATE RELATION BETWEEN PAPERS

The mediate relation between papers is defined as the relations of two paper nodes that share same neighbor node. We consider three types of mediate relations between two papers: they contain same keyword (W_{pp}^{kw}), they were published in same venue (W_{pp}^v) and they were cited by same author (W_{pp}^a). Each element of W_{pp}^{kw}, W_{pp}^v and W_{pp}^a is calculated as:

$$w_{p_ip_j}^{\{kw,v,a\}} = \begin{cases} 1 & \text{if } p_i \text{ shares same node with } p_j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

3) LATENT RELATION BETWEEN PAPERS

Different from other node types, papers contain large amounts of text. These abundant texts contain crucial latent information that can be considered a unique feature to represent a paper. It is obvious that papers with closer relations will have a higher latent correlation, such as semantic and syntax correlation. We use latent Dirichlet allocation (LDA) [26], which is a widely used topic model in many research areas, to excavate the latent correlation hidden between papers. Given a set of documents, the aim of LDA is to explore semantically coherent topics that can be further used to represent the content of documents. Topics can be regarded as better features than words/terms for documents since they embed documents into a lower dimensional space and have good semantic interpretability. We take the paper-topic distribution θ_p as the latent feature for a paper; then, the element of latent relation matrix W_{pp}^l is calculated according to inner product between paper-topic distribution:

$$w_{p_ip_j}^l = \theta_{p_i} \cdot \theta_{p_j} \tag{3}$$

After extracting immediate, mediate and latent relations, we merge these three types of relations to obtain the final paper affinity matrix W_{pp}. It should be noted that W_{pp} is an asymmetric matrix for citing links between papers are directed. Each element of W_{pp} is calculated as:

$$w_{p_ip_j} = min(avg(w_{p_ip_j}^c + w_{p_ip_j}^{kw} + w_{p_ip_j}^v + w_{p_ip_j}^a) + w_{p_ip_j}^l, 1) \tag{4}$$

B. CONSTRUCTION OF FINE-GRAINED AUTHOR AFFINITY MATRIX

Unlike paper nodes, there are no immediate relation between author nodes because all links between authors can be derived through papers in a bibliographic network. Therefore, we only construct mediate and latent relations for authors.

1) MEDIATE RELATION BETWEEN AUTHORS

A scientific paper usually contains several authors, so we consider co-authorship to be the mediate relation between authors. Instead of binary co-authorship, we apply weighted co-authorship [42] to calculate coauthor matrix W_{aa}^c. The reason is that authors should have a higher co-authorship weight if they frequently coauthor, while individual coauthor relationship should be weighted less if the paper has many authors. Each element of W_{aa}^c is calculated as:

$$w_{a_ia_j}^c = \sum_{k=1}^{m} c_{i,j,k}/N_a \tag{5}$$

where $c_{i,j,k}$ denotes author a_i coauthored paper p_k with author a_j and N_a is normalized parameter to ensure the co-authorship weight of an author sums to one.

2) LATENT RELATION BETWEEN AUTHORS

The research activity of authors is always strongly relevant to their latent factors, such as an interest profile and the research community, and an author is more likely to form a link with another author if they have similar latent factors. We apply ACT [43] to extract these latent features for authors. ACT is an extension of LDA that can discover topics among authors in a heterogeneous bibliographic network. We consider these topics as author latent features; then, the element of latent correlation matrix W_{aa}^l between authors is computed as:

$$w_{a_ia_j}^l = \theta_{a_i} \cdot \theta_{a_j} \tag{6}$$

Finally, we merge W_{aa}^c and W_{aa}^l to obtain the final author affinity matrix W_{aa}. Unlike W_{pp}, W_{aa} is a symmetric matrix. Each element of W_{aa} is calculated as:

$$w_{a_ia_j} = min(avg(w_{a_ia_j}^c + w_{a_ia_j}^l), 1) \tag{7}$$

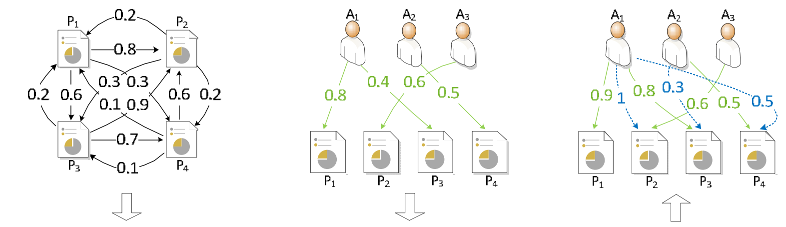
IV. LOW-RANK AND SPARSE MATRIX FACTORIZATION FOR SCIENTIFIC PAPER RECOMMENDATION

In this section, we formally propose LSMFPRec, a novel scientific paper recommendation method that predicts recommendation results from an author citing matrix with fine-grained affinity matrixes extracted from heterogeneous bibliographic network. The detailed parameter learning algorithm is also derived in this section. We believe that LSMFPRec is the first method to use low-rank and sparse matrix factorization for paper recommendation task.

A. RECOMMENDATION WITH BASIC LOW-RANK AND SPARSE FACTORIZATION

Given a rating matrix $X \in \Re^{m\times n}$, the traditional matrix factorization-based CF maps both users and items into a joint latent factor space of low dimensionality such that user-item interactions are modeled as inner products in that space [21]. The resulting dot product between user matrix U and item matrix V captures the recommended rating of users in the characteristics of items, which leads to the following estimate:

$$\underset{U,V}{\arg\min} \ \|X - UV\|_F^2 \tag{8}$$



$$\underset{Y}{\begin{array}{c|cccc} & P_1 & P_2 & P_3 & P_4 \\ \hline P_1 & 1 & 0.2 & 0.2 & 0.1 \\ P_2 & 0.8 & 1 & 0.9 & 0.6 \\ P_3 & 0.6 & 0.3 & 1 & 0.1 \\ P_4 & 0.3 & 0.2 & 0.7 & 1 \end{array}} \times \underset{S}{\begin{array}{c|ccc} & A_1 & A_2 & A_3 \\ \hline P_1 & 0.8 & 0 & 0 \\ P_2 & 0 & 0 & 0.6 \\ P_3 & 0.4 & 0 & 0 \\ P_4 & 0 & 0.5 & 0 \end{array}} = \underset{YS}{\begin{array}{c|ccc} & A_1 & A_2 & A_3 \\ \hline P_1 & 0.9 & 0.1 & 0.1 \\ P_2 & 1 & 0.3 & 0.6 \\ P_3 & 0.8 & 0.1 & 0.2 \\ P_4 & 0.5 & 0.5 & 0.1 \end{array}}$$

FIGURE 3. An example for generating new rating matrix *YS*.

The above matrix factorization manner has two drawbacks when is applied to recommendation. First, the predicted rating is recovered from intermediate matrix, which may cause error propagates from intermediate matrix into final prediction. Second, the manner lacks interpretability due to the uninterpretable low dimensions. Authors working in a similar research field will always read and cite some influential papers when writing a scientific paper. According to this rating pattern, the rating matrix for influential papers presents low-rank character. In addition, the rating pattern for uninfluential papers seems more random. Moreover, compared with influential papers, uninfluential papers are usually less cited. Therefore, the rating matrix for uninfluential papers presents sparse character.

Based on the above low-rank and sparse characteristics in rating activity of scientific work, we model paper recommendation as matrix completion of the author rating matrix $X \in \Re^{m \times n}$ from a low-rank matrix $L \in \Re^{m \times n}$ and a sparse matrix $S \in \Re^{m \times n}$. Although many LSMF methods have been proposed, including RPCA [24], LU_CRTP [25] and GoDec [23], we choose GoDec for its effectiveness and efficiency. By applying GoDec method, the objective function is formulated according to mean-square error (MSE) as follows.

$$\begin{aligned} &\arg\min_{L,S} \ \|X - L - S\|_F^2 \\ &s.t. \ rank\,(L) < r \\ &\qquad card\,(S) < k \end{aligned} \tag{9}$$

where r and k are hyper parameters that represent the rank range of L and the cardinality range of S, respectively.

B. INTEGRATING PAPER AND AUTHOR AFFINITY MATRIXES FOR LSMF

The objective function in above subsection only contains author rating information. In Section 3, we have extracted paper and author affinity matrixes from heterogeneous bibliographic network. We integrate these matrixes into the original LSMF in this section.

1) USING THE PAPER AFFINITY MATRIX

In this subsection, we integrate the paper affinity matrix W_{pp}. For brevity, we denote Y as W_{pp}. In the sparse matrix S in Eq. 9, each of its columns can be considered as the rating of an author for partial papers. If we let the affinity matrix Y multiply S, then we can obtain a new matrix YS. Now, let us carefully examine YS. We can see that YS can be considered as a completion of S using Y. Fig. 3 shows an example of obtaining YS from Y and S. As seen in Fig. 3, S is a sparse matrix in which most values are zero; thus, S can be considered as a bipartite author-paper network with few links. A new author rating matrix YS can be obtained from S by left multiplying Y. If we assume that there is a new link if an element of YS exceeds threshold 0.3, then we obtain 3 new rating relations (with the blue color in the matrix YS and blue dotted line in the related bipartite author-paper network). The results have sufficient interpretability. First, the original links of S (with the green color in matrix S and green line in the related bipartite author-paper network) are preserved in YS. Second, the directed correlations between papers will propagate to author rating. For example, the directed correlation from p_4 to p_2 is 0.6 in Y, and a_2 rated p_4. Thus, it is highly likely that a_2 will cite p_4. Other new author ratings, such as $a_1 \to p_2$ and $a_1 \to p_4$, hold same regularity.

As we can see, YS is a new author rating matrix completed from Y and S. Therefore, it is reasonable that YS holds same low-rank and sparse characteristics as the original author rating matrix X. Thus, we can obtain the following objective function based on YS.

$$\begin{aligned} &\arg\min_{L,S,M} \ \|YS - L - M\|_F^2 \\ &s.t. \ rank(L) < r \\ &\qquad card\,(S) < k \\ &\qquad card\,(M) < q \end{aligned} \tag{10}$$

where $M \in \Re^{m \times n}$ is a sparse matrix with cardinality range q. There are two functions for matrix M. One is that L is a low-rank matrix while YS is not, so we need a non-low-rank matrix

M to make the equation holds. The other is that M can be considered the loss error between YS and L.

2) USING THE AUTHOR AFFINITY MATRIX

While paper affinity matrix Y is used to expand sparse matrix S, we also utilize author affinity matrix W_{aa} to regularize the learning of S. The author affinity matrix W_{aa} contains pairwise linkage information between each two authors, which offers an additional network topology structure constraint. In Eq. 9, each column of S represents the interest of an author towards uninfluential papers, which reflects the randomness of the author's rating pattern. We assume that this randomness also has its own regularities, which is constrained by the network topology structure of author affinity matrix W_{aa}. The reason is that authors with a closer correlation will have higher possibility to co-rate the same papers. Since the network related to W_{aa} is weighted and undirected, we apply graph regularization to model the constraint. The objective function is then obtained as follows:

$$\arg\min_{S} \sum_{i=1}^{m}\sum_{j=1}^{m}(s_i - s_j)^2 W_{a_ia_j} = tr(S^T HS) \tag{11}$$

where $tr(\cdot)$ denotes the trace of a matrix, $H = D - W_{aa}$ denotes the Laplacian matrix, and $D = diag(\sum_j W_{a_ia_j})$.

3) OVERALL OBJECTIVE FUNCTION

Considering the original author rating matrix, the paper affinity matrix and the author affinity matrix all together, we can obtain the overall objective function of LSMFPRec as follows:

$$\begin{aligned}
&\arg\min_{L,S,M} \|X-L-S\|_F^2 + \alpha\,\|YS - L - M\|_F^2 + \beta tr(S^T HS)\\
&\quad s.t.\ rank\ (L) < r,\\
&\qquad card\ (S) < k,\\
&\qquad card\ (M) < q
\end{aligned} \tag{12}$$

where α and β are regularization parameters to control the weight of each term.

C. ESTIMATION PROCESS

In this subsection we formulate the learning process of LSMFPRec. The estimation process applies an alternative optimization scheme that learns each variable separately by fixing others. At each iteration step t, the optimization problem of Eq. 12 is equal to solving the following three subproblems until converging to a local minimum.

$$\begin{cases}
L_t = \arg\min\limits_{rank(L)<r} \|X - L - S_{t-1}\|_F^2 \\
\qquad\qquad + \alpha\,\|YS_{t-1} - L - M_{t-1}\|_F^2) \\
S_t = \arg\min\limits_{card(S)<k} \|X - L_{t-1} - S\|_F^2 \\
\qquad\qquad + \alpha\,\|YS - L_{t-1} - M_{t-1}\|_F^2 + \beta tr(S^T HS) \\
M_t = \arg\min\limits_{card(M)<q} \|YS_{t-1} - L_{t-1} - M\|_F^2
\end{cases} \tag{13}$$

We now derive the update rules for L, S and M during each alternative step t.

1) UPDATE RULE FOR L

The objective function of L in t-th iteration can be rewritten as:

$$\begin{aligned}
J_L &= \|X - L - S_{t-1}\|_F^2 + \alpha\,\|YS_{t-1} - L - M_{t-1}\|_F^2 \\
&= tr[(X - L - S_{t-1})^T (X - L - S_{t-1})] \\
&\quad + \alpha tr[(YS_{t-1} - L - M_{t-1})^T (YS_{t-1} - L - M_{t-1})] \\
&= tr[S_{t-1}^T L + \alpha M_{t-1}^T L - X^T LY^T L - L^T X - \alpha L^T YS_{t-1} \\
&\quad -\alpha S_{t-1}^T + L^T S_{t-1} + \alpha L^T M_{t-1} + L^T L + \alpha L^T L) + C \\
&= tr[(\frac{X + \alpha YS_{t-1} - S_{t-1} - \alpha M_{t-1}}{1+\alpha} - L)^T \\
&\quad \times (\frac{X + \alpha YS_{t-1} - S_{t-1} - \alpha M_{t-1}}{1+\alpha} - L)] + C \\
&= \left\| \frac{X + \alpha YS_{t-1} - S_{t-1} - \alpha M_{t-1}}{1+\alpha} - L) \right\|_F^2
\end{aligned} \tag{14}$$

where C is a constant that is neglected in final equation.

Based on GoDec algorithm, the objective function of Eq. 14 can be solved by updating L_t through singular value hard thresholding as:

$$\begin{aligned}
L_t &= \sum_{i=1}^{r} \lambda_i U_i V_i^T, \\
svd(\frac{X + \alpha YS_{t-1} - S_{t-1} - \alpha M_{t-1}}{1+\alpha}) &= U\Lambda V^T
\end{aligned} \tag{15}$$

where $svd(\cdot)$ denotes singular value decomposition.

2) UPDATE RULE FOR S

Considering the objective function in Eq. 13 in relation to S, the t-th iteration of S can be rewritten as:

$$\begin{aligned}
J_S &= \|X - L_{t-1} - S\|_F^2 + \alpha\,\|YS - L_{t-1} - M_{t-1}\|_F^2 \\
&\quad + \beta tr(S^T HS) \\
&= tr[(X - L_{t-1} - S)^T (X - L_{t-1} - S)] \\
&\quad + \alpha tr[(YS - L_{t-1} - M_{t-1})^T (YS - L_{t-1} - M_{t-1})] \\
&\quad + \beta tr(S^T HS) \\
&= tr[(\alpha Y^T Y + \beta H^T + I) \\
&\quad \times (\frac{X - L_{t-1} + \alpha Y^T L_{t-1} + \alpha Y^T M_{t-1}}{\alpha Y^T Y + \beta H + I} - S) \\
&\quad \times (\frac{X - L_{t-1} + \alpha Y^T L_{t-1} + \alpha Y^T M_{t-1}}{\alpha Y^T Y + \beta H + I} - S)^T] + C \\
&\doteq tr[(\frac{X - L_{t-1} + \alpha Y^T L_{t-1} + \alpha Y^T M_{t-1}}{\alpha Y^T Y + \beta H + I} - S)^T \\
&\quad \times (\frac{X - L_{t-1} + \alpha Y^T L_{t-1} + \alpha Y^T M_{t-1}}{\alpha Y^T Y + \beta H + I} - S)] + C \\
&= \left\| \frac{X - L_{t-1} + \alpha Y^T L_{t-1} + \alpha Y^T M_{t-1}}{\alpha Y^T Y + \beta H + I} - S) \right\|_F^2
\end{aligned} \tag{16}$$

The equivalence between the second and third lines from the bottom needs to be proved, for $tr(AB) \leq tr(A)tr(B)$.

Lemma 1 ((The Second-Derivative Test) [44]): Let $f(X)$: $\Re^{m\times n} \rightarrow \Re$ be a real-valued function defined on a set $X \in \Re^{m\times n}$. Assume that f is twice differentiable at an interior point C of X. If

$$\nabla_C f(C) = \left.\frac{\partial f(X)}{\partial X}\right|_{X=C} = \mathbf{0}_{m\times n} \text{ and} \tag{17}$$

$$\nabla^2_C f(C) = \left.\frac{\partial^2 f(X)}{\partial vec(X)\partial (vecX)^T}\right|_{X=C} \succeq 0 \tag{18}$$

then f has a local minimum at C.

Based on lemma 1, we now provide the following theorem.

Theorem 1: The optimization problem

$$\begin{aligned}\arg\min_S\ & tr[(\alpha Y^T Y + \beta H + I) \\ & \times (\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I} - S)^T \\ & \times (\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I} - S)]\end{aligned} \tag{19}$$

which shares the same local minimum S with the optimization problem.

$$\begin{aligned}\arg\min_S\ & tr[(\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I} - S)^T \\ & \times (\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I} - S)]\end{aligned} \tag{20}$$

Proof: Because the above optimization problems are all related to S, we denote B as

$$B = \frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I} \tag{21}$$

Then, the optimization functions of Eq. 19 and Eq. 20 are rewritten as:

$$J_1 = tr[(\alpha Y^T Y + \beta H + I)(B - S)^T (B - S)] \tag{22}$$

$$J_2 = tr[(B - S)^T (B - S)] \tag{23}$$

First, according to Eq. 17 in Lemma 1, we derive the first derivative of S in J_1 and J_2 as:

$$\frac{\partial J_1(S)}{\partial S} = (\alpha Y^T Y + \beta H + I)(-2B^T + 2S^T) = \mathbf{0} \tag{24}$$

$$\frac{\partial J_2(S)}{\partial S} = -2B^T + 2S^T = \mathbf{0} \tag{25}$$

Then, we obtain the same $S = B$ from the above two equations.

Second, we derive the second derivative of S in J_1 and J_2 according to Eq. 18 in Lemma 1. Because $Y^T Y$ and the Laplacian matrix H are all positive semidefinite matrixes, the following inequalities hold.

$$\frac{\partial^2 J_1(S)}{\partial vec(S)\partial (vecS)^T} = 2(\alpha Y^T Y + \beta H + I) \succeq 0 \tag{26}$$

$$\frac{\partial^2 J_2(S)}{\partial vec(S)\partial (vecS)^T} = 2I \succeq 0 \tag{27}$$

By applying Lemma 1, we can see that the optimization problem in Eq. 19 shares the same local minimum S with the optimization problem in Eq. 20. □

Based on the objective function in Eq. 16 and GoDec, the update rule for S_t can be solved though entry-wise hard thresholding, which is shown as follows:

$$\begin{aligned} S_t &= \Gamma_\Omega(\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I}), \\ \Omega &: \left|(\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I})_{i,j\in\Omega}\right| \neq 0 \\ \text{and} &\geq \left|(\frac{X - L_{t-1} + \alpha L_{t-1} Y + \alpha M_{t-1} Y}{\alpha Y^T Y + \beta H + I})_{i,j\notin\Omega}\right|, \\ |\Omega| &\leq k \end{aligned} \tag{28}$$

where $|A|$ denotes the l_0 norm of A and $\Gamma_\Omega(A)$ denotes the projection matrix to an entry set. Ω is the nonzero entry set of the first k largest entries of A.

3) UPDATE RULE FOR M

The objective function for M is a simple ℓ^2-norm subtraction form, so we can directly obtain the update rule of M as:

$$\begin{aligned} M_t = \Gamma_\Psi(YS_{t-1} - L_{t-1}), \quad & \Psi : \left|(YS_{t-1} - L_{t-1})_{i,j\in\Psi}\right| \neq 0 \\ \text{and} \geq \left|(YS_{t-1} - L_{t-1})_{i,j\notin\Psi}\right|, \quad & |\Psi| \leq q \end{aligned} \tag{29}$$

where Ψ is the nonzero entry set of the first q largest entries of $YS_{t-1} - L_{t-1}$.

D. ACCELERATE LSMFPRec WITH BRP

The parameter estimation of L_t in Eq. 15 uses SVD, which is a time-consuming process. The time complexity of SVD is $\min(mn^2, m^2n)$ flops. When the original rating matrix is too large, the decomposition is impractical. Therefore, we apply BPR to accelerate the parameter estimation of LSMFPRec. More specifically, we apply the power scheme [45] to accelerate the decay of singular values. For brevity, we denote Z as $(X + \alpha YS_{t-1} - S_{t-1} - \alpha M_{t-1})/(1-\alpha)$. Then, a new matrix $\tilde{Z}$ that is to be decomposed can be obtained by:

$$\tilde{Z} = (ZZ^T)^b Z \tag{30}$$

where b is a power scheme term.

It should be noted that both Z and $\tilde{Z}$ share the same singular vectors [8]. Thus, the r rank approximation of $\tilde{Z}$ via BPR is:

$$\tilde{L} = F_1(A_2^T F_1)^{-1} F_2^T \tag{31}$$

where $F_1 = \tilde{Z}A_1$ and $F_2 = \tilde{Z}^T A_2$. $A_1 \in \Re^{n\times r}$ and $A_2 \in \Re^{m\times r}$ are independent random matrixes built from F_1 and F_2 by subsampled randomized Fourier transform (SRFT) [46] with Gaussian noise, respectively.

To acquire the r rank approximation of $\tilde{Z}$, we apply QR factorization of F_1 and F_2. Then, the low-rank approximation of $\tilde{Z}$ is given by:

$$L = (\tilde{L})^{1/(2b+1)} = Q_1[R_1(A_2^T F_1)^{-1} R_2^T]^{1/(2b+1)} Q_2^T \tag{32}$$

where $Q_1 R_1 = F_1$ and $Q_2 R_2 = F_2$ represent the QR factorization of F_1 and F_2, respectively. The value of power scheme term b is set to 2, as in GoDec.

By applying the above process, we can reduce the time complexity of SVD in Eq. 15 from $\min(mn^2, m^2n)$ flops to

Algorithm 1 Estimation Process of LSMFPRec

Input: author rating matrix X, author affinity matrix W_{aa}, paper affinity matrix Y, rank range r, cardinality range k and q, threshold ε, bias term α and β.
Output: low-rank matrix L, sparse matrix S
Initialize: $L_0 := X, S_0 := 0, M_0 := 0, t := 0$
1: **while** the loss error of Eq. 12 $> \varepsilon$ **do**
2: $t := t + 1$;
3: $Z = (X + \alpha YS_{t-1} - S_{t-1} - \alpha M_{t-1})/(1 - \alpha)$;
4: $\tilde{L} = (ZZ^T)^b Z$;
5: $F_1 = \tilde{L}A_1, A_2 = F_1$;
6: $F_2 = \tilde{L}^T F_1 = Q_2R_2, F_1 = \tilde{L}F_2 = Q_1R_1$;
7: **if** $rank(A_2^T F_1) < r$
8: $r := rank(A_2^T F_1)$, go to step (1);
9: **end if**
10: update L_t according to Eq. 32;
11: update S_t according to Eq. 28;
12: update M_t according to Eq. 29;
13: **end while**

$\min(r^2m, r^2n, mnr)$ flops. Because r is usually much smaller than m and n, the above process can significantly reduce the time complexity of LSMFPRec.

E. LEARNING ALGORITHM AND RECOMMENDATION PROCESS OF LSMFPRec

The learning algorithm of LSMFPRec is summarized in Algorithm 1. In steps 10–12, the algorithm updates L, S and M iteratively until convergence, and the optimal solution of the overall objective function (Eq. 12) can be obtained simultaneously.

After the estimation process, we can obtain the approximate author rating matrix $\tilde{X}$ from the learned low-rank matrix L and sparse matrix S as:

$$\tilde{X} = L + S \tag{33}$$

Then, we take the top n entries in a column of matrix $\tilde{X}$, which are zero in the original matrix X, as the recommended results for an author.

F. CONVERGENCE ANALYSIS OF LSMFPRec

We now analyze the convergence property of LSMFPRec. First, let us introduce following lemmas:

Lemma 2 [39]: Given a real matrix $Z \in \Re^{m\times n}$ with singular value decomposition $Z = U\Lambda V^T = U_1\Lambda_1 V_1^T + U_2\Lambda_2 V_2^T$, the Eq. 32 approximates Z with the error upper bounded by

$$\|Z - L\|_F^2 \le \left(\left\| \Lambda_2^{2(2b+1)}(V_2^T A_1)(V_1^T A_1)^T \Lambda_1^{-(2b+1)} \right\|_F^2 + \left\| \Lambda_2^{2b+1} \right\|_F^2\right)^{1/(2b+1)} \tag{34}$$

Lemma 3 [47]: Given a minimization problem

$$\min \|f - u\|_F^2 + \Theta(u) \tag{35}$$

where $\Theta(u)$ is a sparsity constraint in the form of l^p-penalties that is convex but possibly non-smooth. Exclusively using iterated hard thresholding produces a sequence $\{u_n\}$ that converges linearly to the unique minimizer u^*.

Based on the above two lemmas, we now provide the following theorem.

Theorem 2: The iterations of algorithm 1 lead the objective function of Eq. 12 converges to a local minimum.

Proof: Algorithm 1 is equivalent to solve three sub-problems in Eq. 13. Let us denote E_t^1, E_t^2 and E_t^3 as the overall loss error of these three sub-problems at t-th iterations. More specifically,

$$\begin{cases} E_t^1 = \|X - L_t - S_{t-1}\|_F^2 + \alpha\,\|YS_{t-1} - L_t - M_{t-1}\|_F^2 \\ \quad +\beta tr(S_{t-1}^T HS_{t-1}) \\ E_t^2 = \|X - L_t - S_t\|_F^2 + \alpha\,\|YS_t - L_t - M_{t-1}\|_F^2 \\ \quad +\beta tr(S_t^T HS_t) \\ E_t^3 = \|X - L_t - S_t\|_F^2 + \alpha\,\|YS_t - L_t - M_t\|_F^2 \\ \quad +\beta tr(S_t^T HS_t) \end{cases} \tag{36}$$

By applying Lemma 1, we can obtain $E_{t-1}^3 \ge E_t^1$. We can also obtain $E_t^1 \ge E_t^2$ and $E_t^2 \ge E_t^3$ by applying Lemma 2. Therefore, the loss errors in Eq. 13 are maintained as descending throughout the iterations of algorithm 1 as:

$$E_1^1 \ge E_1^2 \ge E_1^3 \ge E_2^1 \ge \ldots \ge E_t^1 \ge E_t^2 \ge E_t^3 \ge E_{t+1}^1 \ge \ldots \tag{37}$$

It can be seen that the overall loss error of Eq. 12 decreases monotonically. Therefore, algorithm 1 produces a sequence that leads the objective function of Eq. 12 converges to a local minimum. □

V. EXPERIMENTS AND EVALUATIONS

A. DATASETS AND PREPROCESSING

We choose two different real-world datasets to validate the effectiveness of LSMFPRec. One is the ACL anthology network (AAN), which is a bibliographic dataset. The other is the scholarly social network CiteULike.

·AAN dataset[3]: Radev *et al.* [48] established the ACL Anthology Network (AAN) dataset that contains full-text information of conference and journal papers in computational linguistics and natural language processing area. We used a subset of the 2013 release, which contains papers published from 1965 to 2013. We removed papers with incomplete information, e.g., missing authors or keywords. We also removed authors who cited fewer than 10 papers. The final dataset contains 4,497 authors and 12,274 papers with 187,540 observed author-paper pairs. The sparsity of the rating matrix is 99.66%. On average, each author cited 42 papers in the data set, ranging from 10 to 1145, and 86.68% of authors cited fewer than 100 papers.

[3] clair.eecs.umich.edu/aan/

TABLE 1. Statistics of Aan and Citeulike.

	Authors	Papers	Rating	Sparsity
AAN	4,497	12,274	187,540	99.66%
CiteULike	5,551	16,980	204,987	99.78%

·CiteULike dataset[4]: CiteULike is a well-known scholarly social network that allows researchers to store, organize, share and discover links to academic research papers. When a researcher posts a paper, CiteULike will automatically extract its abstract, title and keywords from the Web. We use the dataset released in [13], which contains 5,551 authors and 16,980 papers. The overall number of ratings is 204,997. The sparsity of the rating matrix is 99.78%. It should be noted that the rating values in CiteULike are all binary. If a user reads or posts a paper, the corresponding rating is 1. Otherwise, the corresponding rating is 0. Similar to [13], we removed the authors with fewer than three papers.

The statistics of these two datasets are summarized in Table 1. We can see that both datasets are extremely sparse. The ratios of rated entries (equal to 1) in the rating matrixes of AAN and CiteULike are 0.0034 and 0.0022, respectively. To extract topics using LDA and ACT, the titles and abstracts are extracted as text content for each paper. Then, we used Porter stemmer to remove stop words and extract stems. Words that is consist of fewer than three characters and appear fewer than ten times are also removed to reduce the impact of short words.

B. EVALUATION METHODOLOGY

For performance evaluation, we randomly sampled 10% of rated entries as test set. The rest of ratings are merged as training set. We first used training set to obtain an estimated low rank matrix and sparse matrix and then presented the top N papers to authors who pertained to test set according to approximated rating matrix. To reduce the errors caused by inappropriate sampling, the experiments were cross validated on 10 sets of randomly chosen samples.

Following common practice in information retrieval (IR) task, we employed following two evaluation metrics to evaluate recommendation results:

· Recall is a commonly used metric for IR field. Recall@N measures the rate of real ratings that are retrieved in the top N recommendation list. This metric is calculated as follows:

$$Recall = \frac{\sum_{a\in Q(A)} |R(a) \cap T(a)|}{\sum_{a\in Q(A)} |T(a)|} \tag{38}$$

where $Q(A)$ is testing author set, $T(a)$ is ground truth papers rated by author a, and $R(a)$ is recommended papers for author a.

· Normalized Discounted Cumulative Gain (NDCG) is a retrieval metric that was devised specifically for measuring ranking in IR tasks. For an author a, the ranked recommendations are examined from the top-ranked down. NDCG is computed as:

$$NDCG_a = n_a \sum_j (2^{r(j)} - 1)/\log(1+j) \tag{39}$$

where n_a is a normalization constant chosen such that a perfect ordering would obtain $NDCG_a = 1$, and $r(j)$ is integer label for the relevance level of j-th paper in sorted recommendation list. NDCG is well-suited for paper recommendation evaluation, as it rewards relevant papers in the top-ranked results more heavily than those ranked lower.

C. COMPARISON WITH OTHER APPROACHES

To evaluate the recommendation performance of LSMFPRec, we compare it against several other approaches. The compared methods are summarized as follows:

· CF [21]: This is the basic model-based CF that is based on traditional matrix factorization. Without loss of generality, we do not add any biases in factorization process.

· CTR [13]: CTR combines collaborative filtering and probabilistic topic modeling, which can learn latent matrixes and topics simultaneously. We employ in-matrix prediction in CTR since our recommendation scenario belongs to traditional collaborative filtering.

· RCTR [15]: RCTR extends CTR by adding user-item feedback information, item-content information, and network structure among items. It also provides a family of link probability functions to increase the capacity of model. Similar to CTR, we employ RCTR for only in-matrix prediction.

· LRCBF [57]: LRCBF considers papers that marked as interesting by users are positive instances, while other papers published at same venue are negative instances. Then LRCBF uses these instance pairs to train Rank SVM [60] to perform recommendation. We use weighting based validation methods in experiments for it is better than pruning based validation according to [57].

· ConceptPRec [58]: ConceptPRec uses Paragraph Vector [59] to learn deep representations of papers. We concatenate user rated papers into an input paper, and recommend papers that have higher similarity with the input paper.

· SLIM [40]: SLIM learns a user latent matrix and a sparse item coefficient matrix from the original rating by matrix factorization. Despite different definition on the latent item matrix, SLIM still applies nearly the same recommendation approach as traditional CF.

· LSMC [27]: LSMC estimates only a single low-rank and sparse matrix from the original rating matrix. Without any further computation, the final predicted rating can be directly obtained after learning process of LSMC.

In addition to the above approaches, we also compare the performance of LSMFPRec by using only partial link information. LSMFPRec-b denotes basic LSMFPRec that does not utilize paper and author affinity matrixes, and its objective function is equal to Eq. 9. It should be noted that LSMFPRec degenerates into original GoDec in this case.

[4]www.citeulike.org

TABLE 2. Performance comparison between different methods on AAN.

Method	NDCG						Recall					
	50	100	150	200	250	300	50	100	150	200	250	300
CF	0.1353	0.1680	0.2054	0.2356	0.2572	0.2659	0.0881	0.1158	0.1392	0.1491	0.1531	0.1595
CTR	0.1851	0.2523	0.3177	0.3680	0.3993	0.4198	0.1142	0.1462	0.1752	0.1893	0.1992	0.2051
RCTR	0.2046	0.3221	0.3952	0.4541	0.4907	0.5031	0.1531	0.2054	0.2460	0.2684	0.2865	0.3053
LRCBF	0.1954	0.2295	0.2973	0.3560	0.3742	0.3971	0.1293	0.1693	0.2048	0.2364	0.2551	0.2930
ConceptPRec	0.1402	0.1729	0.2590	0.3247	0.3490	0.3701	0.0904	0.1327	0.1690	0.1803	0.1924	0.2037
SLIM	0.1541	0.2047	0.2462	0.2792	0.3102	0.3159	0.0941	0.1293	0.1482	0.1641	0.1757	0.1841
LSMC	0.1721	0.2331	0.2780	0.3176	0.3442	0.3597	0.1093	0.1321	0.1626	0.1779	0.1886	0.1946
LSMFPRec-b	0.1690	0.2489	0.2954	0.3390	0.3663	0.3853	0.1083	0.1492	0.1655	0.1797	0.1925	0.1953
LSMFPRec-p	0.1935	0.2905	0.3593	0.4154	0.4535	0.4681	0.1254	0.1598	0.1974	0.2185	0.2391	0.2591
LSMFPRec-a	0.1842	0.2702	0.3352	0.3883	0.4241	0.4453	0.1138	0.1552	0.1783	0.2079	0.2251	0.2464
LSMFPRec	0.2292	0.3441	0.4265	0.4843	0.5265	0.5393	0.1698	0.2241	0.2785	0.3052	0.3128	0.3385

TABLE 3. Performance comparison between different methods on CiteULike.

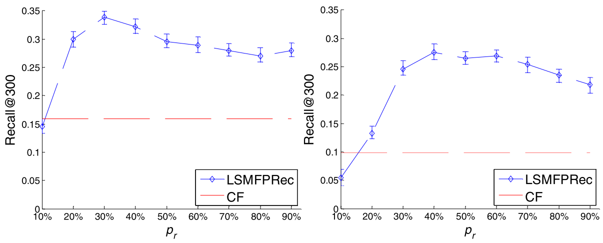
Method	NDCG						Recall					
	50	100	150	200	250	300	50	100	150	200	250	300
CF	0.1084	0.1450	0.1690	0.1883	0.2065	0.2230	0.0513	0.0731	0.0835	0.0895	0.0941	0.0993
CTR	0.1458	0.2052	0.2590	0.2991	0.3259	0.3441	0.0864	0.1194	0.1429	0.1641	0.1845	0.1953
RCTR	0.1565	0.2431	0.2921	0.3335	0.3592	0.3770	0.0986	0.1562	0.1938	0.2292	0.2461	0.2561
LRCBF	0.1568	0.2399	0.2603	0.3185	0.3427	0.3601	0.0925	0.1302	0.1658	0.2047	0.2219	0.2283
ConceptPRec	0.1242	0.1590	0.2252	0.2703	0.2998	0.3275	0.0799	0.1063	0.1406	0.1591	0.1736	0.1847
SLIM	0.1159	0.1609	0.1932	0.2232	0.2392	0.2490	0.0654	0.0893	0.1154	0.1264	0.1366	0.1412
LSMC	0.1398	0.1862	0.2190	0.2567	0.2841	0.2939	0.0751	0.1053	0.1304	0.1531	0.1684	0.1741
LSMFPRec-b	0.1302	0.1954	0.2396	0.2804	0.2973	0.3082	0.0732	0.1141	0.1381	0.1635	0.1764	0.1845
LSMFPRec-p	0.1540	0.2302	0.2752	0.3219	0.3502	0.3658	0.0894	0.1293	0.1625	0.1931	0.2145	0.2309
LSMFPRec-a	0.1495	0.2198	0.2601	0.3154	0.3429	0.3582	0.0827	0.1154	0.1442	0.1821	0.2064	0.2158
LSMFPRec	0.1651	0.2694	0.3265	0.3662	0.3851	0.3941	0.1031	0.1741	0.2190	0.2452	0.2642	0.2752

We denote LSMFPRec-p and LSMFPRec-a as learning only by using paper affinity matrix and author affinity matrix, respectively.

The performance of the above methods in two datasets are shown in Table 2 and 3. It is obvious that our LSMFPRec leads the performance in all cases. The more detailed analyses are outlined as follows:

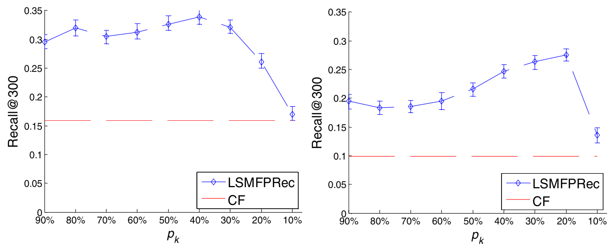
We first analyze the performance among CF, CTR, RCTR, LRCBF, ConceptPRec and LSMFPRec. Due to the lack of content information, we can see that CF is obviously worse than any of the other methods in terms of all metrics. Although ConceptPRec recommends papers uses deep learning, the performance of ConceptPRec is just merely better than CF. We believe the reason is that ConceptPRec neglects various link information when learning deep representations of papers. CTR works much better than CF, which indicates that content information plays an important role in recommending scientific papers. LRCBF performs better than CTR for it explores the relationships of papers in same venue. RCTR shows a clear performance gain over LRCBF because it explores more sophisticated link information when extracting topics of content. Compared with RCTR, the average performance improvements of LSMFPRec for NDCG in AAN and CiteUlike are 7.99% and 11.17%. For recall, LSMFPRec achieves 11.17% and 8.47% higher performance than RCTR does in AAN and CiteUlike. The experimental results demonstrated that our LSMFPRec completely and significantly exceeds RCTR in all metrics. The reasons are twofolds: First, although both RCTR and LSMFPRec utilize homogeneous link information of paper and authors, the links used in LSMFPRec are all fine-grained. We constructed paper and author affinity matrixes by diversifying links and exploiting topic correlations, which can generate highly qualified paper-paper and author-author links to improve performance. Second, the recommendation results of RCTR are generated by the inner product of a latent user vector and an item vector, while our LSMFPRec does not have such intermediate outputs. This character reduces the error accumulated in generating interim outputs.

Next, we analyze the results among SLIM, LSMC and LSMFPRec. It is not surprising that SLIM and LSMC perform much worse than LSMFPRec, because both of them neglect content information. In particular, LSMFPRec achieves 73.95% higher recall@300 than LSMC in AAN and 94.90% higher recall@300 than SLIM in CiteULike. Although SLIM explores sparse correlations between papers,



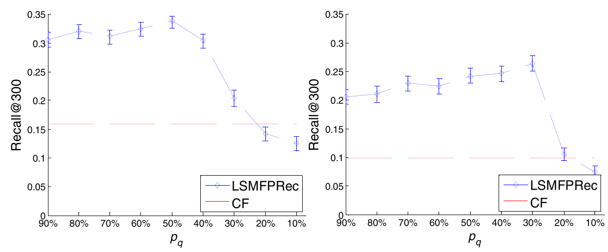
(a) (b)

FIGURE 4. **The performance impact of *r*. (a) AAN dataset. (b) CiteULike dataset.**



(a) (b)

FIGURE 5. **The performance impact of *k*. (a) AAN dataset. (b) CiteULike dataset.**



(a) (b)

FIGURE 6. **The performance impact of *q*. (a) AAN dataset. (b) CiteULike dataset.**

it is still based on traditional CF, which cannot avoid the defect of intermediate outputs. LSMC improved SLIM by factorizing the original rating matrix directly into a single low-rank and sparse predicted rating matrix, which can prevent all intermediate outputs. However, a single low-rank and sparse rating matrix might be too coarse to match rating patterns and recommend correct papers for researchers.

Finally, we compare the recommendation performance among the variants of LSMFPRec. We can see that without content information, LSMFPRec-b achieves the lowest performance than other variants. LSMFPRec-p works better than LSMFPRec-a, which indicates that paper affinity matrix is more important than author affinity matrix in recommending scientific papers.

D. PARAMETER TUNING

In our method, there are five essential parameters: a rank range r, two cardinality ranges k and q, and two bias terms α and β. In this subsection, the effect of these hyper parameters are studied and evaluated. We evaluate these parameters by empirically fixing others. Due to page limitations, we only demonstrate the tuning results on recall@300; other metrics generate similar results in our experiments.

We first evaluate the effect of r, k and q by empirically fixing $\alpha = 80$ and $\beta = 60$. Since these three parameters are scalars that ranged according to original rating, we transmit them into the following percentage variables for ease of tuning.

$$p_r = \frac{r}{Rank(X)}, \quad p_k = \frac{k}{Size(X)}, \; p_q = \frac{q}{Size(X)} \tag{40}$$

Fig. 4 illustrates the recall@300 by varying r on AAN and CiteULike datasets. Both k and q are empirically set to $40\% \cdot Size(X)$. We can see that when r is very small (about $< 15\% \cdot Rank(X)$), the performance of LSMFPRec is even lower than that of traditional CF. The recommendation is improved as the value of r increases. Our model achieves the best performance when $r = 30\% \cdot Rank(X)$ and $r = 40\% \cdot Rank(X)$ in AAN and CiteULike, respectively.

The cardinality range k and q control the sparsity of S and M, respectively. The recall@300 with varying k and q on the AAN and CiteULike dataset are shown in Fig. 5 and Fig. 6. For AAN dataset, the best performance is achieved when $k = 40\% \cdot Size(X)$ and $q = 50\% \cdot Size(X)$. For CiteULike dataset, the best performance is achieved at $k = 20\% \cdot Size(X)$ and $q = 30\% \cdot Size(X)$. It should be noted that when M is too sparse ($q < 23\% \cdot Size(X)$ in AAN and $q < 20\% \cdot Size(X)$ in CiteULike), LSMFPRec works even worse than traditional CF. The reason is that although M does not participate in final recommendation, an extreme sparse M will produce less noise in decomposition process, which causes overfitting problem. To illustrate the sparsity better, we plot a partial matrix of S with different cardinality ranges k on the CiteULike dataset in Fig. 7, where white points represent zero entries.

We then study the effectiveness of α and β. The two parameters control the importance of paper affinity matrix and author affinity matrix separately. Fig. 8 and Fig. 9 show how our model performs when these two parameters vary on AAN and CiteULike datasets. To investigate further, we also plotted recall contours. We can see that when $\alpha = 0$ and $\beta = 0$, the performance is obviously not satisfactory since no content information is involved. The performance improves when increasing α and β. It should be noted that there is a region where the optimal values of α and β ensure the best prediction accuracy. The region is approximately at $\alpha = 80 \sim 90$ and $\beta = 65 \sim 75$. Moreover, the results also indicate that paper link information plays a more important role than author link information for scientific paper recommendation.

E. RECOMMENDATION ANALYSIS FOR INFLUENTIAL AND UNINFLUENTIAL PAPERS

The predicted results of LSMFPRec are obtained from a low-rank matrix and a sparse matrix. These two matrixes have a good recommendation interpretation on influential and uninfluential papers separately. Although many metrics have

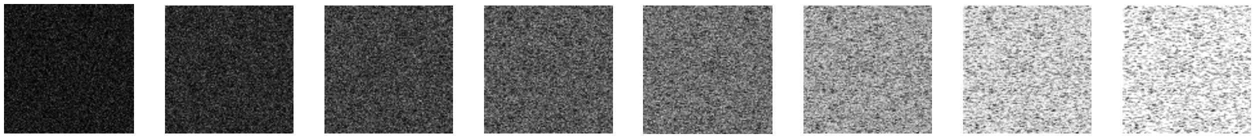


FIGURE 7. Partial of sparse matrix S generated in different cardinality ranges k on CiteULike. The values of p_k are from 0.9 to 0.2, with a 0.1 decrement from left to right.

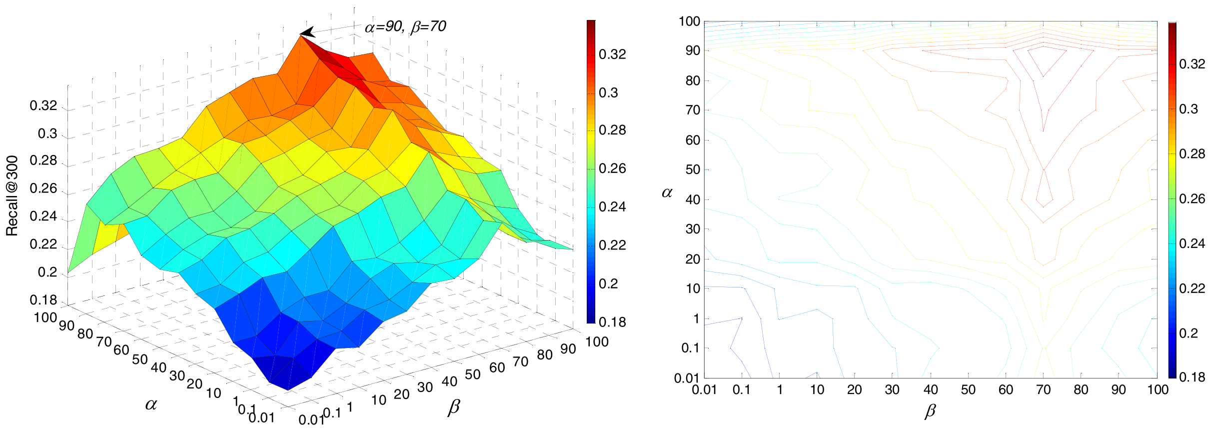


FIGURE 8. The performance impact of LSMFPRec by varying the bias terms α and β in AAN.

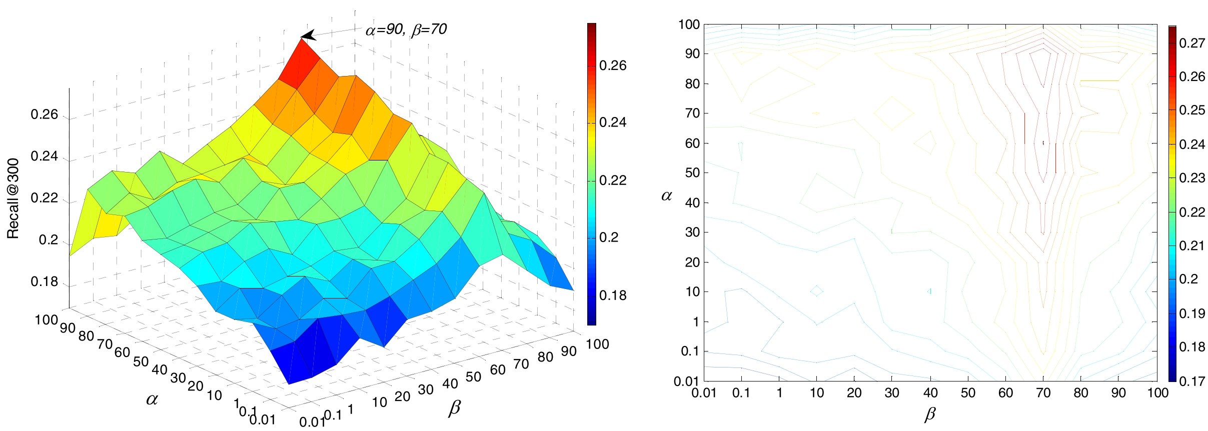


FIGURE 9. The performance impact of LSMFPRec by varying bias term α and β in CiteULike.

TABLE 4. Recommendation results for "Agirre, Eneko".

	Title	Citation count	Contributed matrix
1)	Verb Semantics And Lexical Selection	3170	L
2)	Efficient Third-Order Dependency Parsers	222	L
3)	Indexing With WordNet Synsets Can Improve Text Retrieval	495	L
4)	Unsupervised Word Sense Disambiguation Rivaling Supervised Methods	2292	L
5)	Learning Class-To-Class Selectional Preferences	113	S
6)	Combining Trigram-Based And Feature-Based Methods For Context-Sensitive Spelling Correction	181	S
7)	Constructing Lexical Transducers	155	S
8)	Combining Unsupervised Lexical Knowledge Methods For Word Sense Disambiguation	135	S
9)	Using Machine Translation Evaluation Techniques to Determine Sentence-level Semantic Equivalence	86	S
10)	A Taxonomy For English Nouns And Verbs	171	S

been proposed to measure the degree of influence of scientific papers, we choose citation count without loss of generality. To gain a better insight into LSMFPRec, we present the top 10 correctly recommended articles for an author whose name is "Agirre, Eneko" in AAN dataset, and their citation counts which are extracted from Google Scholar. As shown

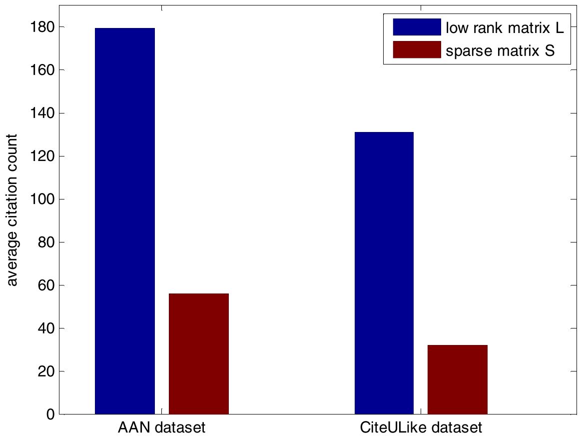


FIGURE 10. **The average citation counts of top 20 recommended results in the low-rank matrix and the sparse matrix.**

in Table 4, the recommended papers contributed by low-rank matrix L are all highly cited (all above 200), which means that L captures the interests of the author on influential papers. In contrast, the citation counts of papers obtained by sparse matrix S are much lower (all below 200), which illustrates that S provides the author with preferred uninfluential papers. We only show the results of one author here, there are more evidences that support the finding in testing set.

To better illustrate the effect of low-rank matrix and sparse matrix, we separately counted the average citation counts of top 20 recommended results in low-rank matrix and sparse matrix. It can be seen in Fig. 10 that the average citation counts of low-rank matrix L are 179 and 131 for AAN and CiteUlike, respectively. In contrast, for sparse matrix, the average citation counts are 56 and 32. The statistical results demonstrated that LSMFPRec can reveal the rating characteristics of influential and uninfluential papers.

VI. CONCLUSIONS

In this paper, we proposed a novel method, named LSMFPRec, for passive paper recommendation. To fully utilize content information and diversified links, we first extracted fine-grained paper and author affinity matrixes from heterogeneous bibliographic network. Then, we seamlessly integrated these fine-grained affinity matrixes into the decomposition process of low-rank and sparse matrix factorization. The estimated low-rank and sparse matrixes are used to generate predicted ratings for authors. LSMFPRec can utilize author rating information, paper content information and network structure to alleviate sparsity and cold start problem encountered by traditional collaborative filtering method. Finally, extensive experiments were conducted on two real-world datasets, AAN and CiteULike, to evaluate the performance. The experimental results demonstrated that our LSMFPRec outperforms other baseline algorithms. Moreover, we demonstrated that LSMFPRec has the ability to reveal the rating characteristics of influential and uninfluential papers.

The matrix decomposition manner of LSMFPRec is sufficiently flexible to integrate other related networks. For example, we can integrate links of traditional social networks into LSMFPRec by adding more matrix decompositions. We can also combine LSMFPRec with impact propagation to better capture influential papers. Moreover, it is easy to design distributed factorization algorithms for LSMFPRec, which would make LSMFPRec scalable for big data modeling. The above possible extensions will be pursued in our future work.

REFERENCES

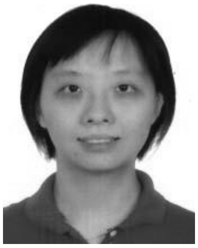
[1] Y. Guo, M. Wang, and X. Li, "Application of an improved Apriori algorithm in a mobile e-commerce recommendation system," *Ind. Manage. Data Syst.*, vol. 117, no. 2, pp. 287–303, 2017.
[2] H. T. H. Nguyen, M. Wistuba, and L. Schmidt-Thieme, "Personalized tag recommendation for images using deep transfer learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2017, pp. 705–720.
[3] F. Yu, N. Che, Z. Li, K. Li, and S. Jiang, "Friend recommendation considering preference coverage in location-based social networks," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2017, pp. 91–105.
[4] S. Deng, L. Huang, and G. Xu, "Social network-based service recommendation with trust enhancement," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8075–8084, Dec. 2014.
[5] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based Web service recommendation," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 453–466, May/Jun. 2015.
[6] F. Meng *et al.*, "A unified graph model for personalized query-oriented reference paper recommendation," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 1509–1512.
[7] L. Guo, X. Cai, F. Hao, D. Mu, C. Fang, and L. Yang, "Exploiting fine-grained co-authorship for personalized citation recommendation," *IEEE Access*, vol. 5, pp. 12714–12725, 2017.
[8] D. Mu, L. Guo, X. Cai, and F. Hao, "Query-focused personalized citation recommendation with mutually reinforced ranking," *IEEE Access*, vol. 6, pp. 3107–3119, 2018.
[9] S. M. McNee *et al.*, "On the recommending of citations for research papers," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 2002, pp. 116–125.
[10] K. D. Bollacker, S. Lawrence, and C. L. Giles, "CiteSeer: An autonomous Web agent for automatic retrieval and identification of interesting publications," in *Proc. ACM 2nd Int. Conf. Auto. Agents*, 1998, pp. 116–123.
[11] R. Torres *et al.*, "Enhancing digital libraries with TechLens+," in *Proc. 4th ACM/IEEE-CS Joint Conf. Digit. Libraries*, 2004, pp. 228–236.
[12] K. Sugiyama and M. Y. Kan, "Scholarly paper recommendation via user's recent research interests," in *Proc. ACM 10th Annu. Joint Conf. Digit. Libraries*, 2010, pp. 29–38.
[13] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 448–456.
[14] S. Purushotham, Y. Liu, and C.-C. J. Kuo, "Collaborative topic regression with social matrix factorization for recommendation systems," in *Proc. Int. Conf. Mach. Learn.*, Madison, WI, USA, 2012, pp. 691–698.
[15] H. Wang and W. J. Li, "Relational collaborative topic regression for recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1343–1355, May 2015.
[16] T. Dai, L. Zhu, X. Cai, S. Pan, and S. Yuan, "Explore semantic topics and author communities for citation recommendation in bipartite bibliographic network," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 4, pp. 957–975, 2018.
[17] 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, Oct. 2015.
[18] M. Gori and A. Pucci, "Research paper recommender systems: A random-walk based approach," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Dec. 2006, pp. 778–781.
[19] X. Ren *et al.*, "Cluscite: Effective citation recommendation by information network-based clustering," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 821–830.

[20] Y.-C. Lee, "Recommendation of research papers in DBpia: A hybrid approach exploiting content and collaborative data," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2016, pp. 2966–2971.
[21] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[22] DM. Pennock, "Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach," in *Proc. 6th Conf. Uncertainty Artif. Intell.* Burlington, MA, USA: Morgan Kaufmann, 2000, pp. 473–480.
[23] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *Proc. ICML*, Bellevue, WA, USA, 2011, pp. 33–40.
[24] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.
[25] L. Grigori, S. Cayrols, and J. W. Demmel, "Low rank approximation of a sparse matrix based on LU factorization with column and row tournament pivoting," *SIAM J. Sci. Comput.*, vol. 40, no. 2, pp. 181–209, 2018.
[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[27] Z.-L. Zhao, "Low-rank and sparse matrix completion for recommendation," in *Proc. ICONIP*. Cham, Switzerland: Springer, 2017, pp. 3–13.
[28] C. Yang *et al.*, "CARES: A ranking-oriented CADAL recommender system," in *Proc. JCDL*, 2009, pp. 203–212.
[29] Z. Cailan, C. Kai, and L. Shasha, "Improved PageRank algorithm based on feedback of user clicks," in *Proc. IEEE CSSS*, Jun. 2011, pp. 3949–3952.
[30] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3419–3430, May 2011.
[31] Y. Mu, J. Dong, X. Yuan, and S. Yan, "Accelerated low-rank visual recovery by random projection," in *Proc. IEEE CVPR*, Jun. 2011, pp. 2609–2616.
[32] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform invariant low-rank textures," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 1–24, 2012.
[33] F. Cao, M. Cai, and Y. Tan, "Image interpolation via low-rank matrix completion and recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1261–1270, Aug. 2015.
[34] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.
[35] H. Yin, "A joint sparse and low-rank decomposition for pansharpening of multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3545–3557, Jun. 2017.
[36] R. S. Rezende, J. Zepeda, J. Ponce, F. Bach, and P. Pérez, "Kernel square-loss exemplar machines for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7263–7271.
[37] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, 2015.
[38] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.
[39] T. Zhou and D. Tao, "Bilateral random projections," in *Proc. IEEE ISIT*, Jul. 2012, pp. 1286–1290.
[40] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-N recommender systems," in *Proc. ICDM*, Dec. 2011, pp. 497–506.
[41] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.
[42] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel, "Co-authorship networks in the digital library research community," *Inf. Process. Manage., Int. J.*, vol. 41, no. 6, pp. 1462–1480, 2005.
[43] J. Tang, R. Jin, and J. Zhang, "A topic modeling approach and its integration into the random walk framework for academic search," in *Proc. ICDM*, Dec. 2008, pp. 1055–1060.
[44] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus With Applications in Statistics and Econometrics* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 1988.
[45] S. T. Roweis, "EM algorithms for PCA and SPCA," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 10, 1997, pp. 626–632.
[46] V. Rokhlin and M. Tygert, "A fast randomized algorithm for overdetermined linear least-squares regression," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 36, pp. 13212–13217, 2008.
[47] K. Bredies and D. Lorenz, "Iterated hard shrinkage for minimization problems with sparsity constraints," *SIAM J. Sci. Comput.*, vol. 30, no. 2, pp. 657–683, Jan. 2008.
[48] D. R. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara "The ACL anthology network corpus," *Lang. Resour. Eval.*, vol. 47, no. 4, pp. 919–944, 2013.
[49] L. Yang, X. Cai, S. Pan, H. Dai, and D. Mu, "Multi-document summarization based on cluster using non-negative matrix factorization," *J. Intell. Fuzzy Syst.*, vol. 33, no. 3, pp. 1867–1879, 2017.
[50] X. Cai, J. Han, S. Pan, and L. Yang, "Heterogeneous information network embedding based personalized query-focused astronomy reference paper recommendation," *Int. J. Comput. Intell. Syst.*, vol. 11, no. 1, pp. 591–599, 2018.
[51] S. Pan *et al.* (2018). "Adversarially regularized graph autoencoder." [Online]. Available: https://arxiv.org/abs/1802.04407
[52] S. Pan *et al.*, "Tri-party deep network representation," *Network*, vol. 11, no. 9, pp. 1895–1901, 2016.
[53] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 744–758, Mar. 2017.
[54] S. Pan, J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Joint structure feature exploration and regularization for multi-task graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 715–728, Mar. 2016.
[55] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1235–1244.
[56] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[57] A. Alzoghbi, V. A. A. Ayala, P. M. Fischer, and G. Lausen, "Learning-to-rank in research paper CBF recommendation: Leveraging irrelevant papers," in *Proc. 3rd Workshop New Trends Content-Based Recommender Syst.*, 2016, pp. 43–46.
[58] R. Sharma, D. Gopalani, and Y. Meena, "Concept-based approach for research paper recommendation," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.* Cham, Switzerland: Springer, 2017, pp. 687–692.
[59] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. II-1188–II-1196.
[60] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*. Cambridge, MA, USA: MIT Press, 2000, ch. 1, pp. 115–132.

TAO DAI received the M.S. degree in software engineering from Xi'an Jiaotong University, China, in 2011, where he is currently pursuing the Ph.D. degree with the School of Software Engineering. His main research interests include machine learning and information retrieval.

TIANYU GAO received the B.S. degree in computer science from Shanghai Jiaotong University, China, in 2016. He is currently pursuing the M.S. degree with the School of Software Engineering, Xi'an Jiaotong University, China. His main research interests include machine learning and information retrieval.

LI ZHU received the Ph.D. degree in computer system architecture from Xi'an Jiaotong University, China, in 2000. He is currently an Associate Professor with the School of Software Engineering, Xi'an Jiaotong University. His research interests include machine learning and computer networking.



XIAOYAN CAI received the Ph.D. degree from Northwestern Polytechnical University, China, in 2009. She was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. She is currently an Associate Professor with the School of Automation, Northwestern Polytechnical University. Her current research interests include document summarization, information retrieval, and machine learning.

SHIRUI PAN received the Ph.D. degree in computer science from the University of Technology Sydney (UTS), Australia, in 2015. He is currently a Research Associate with the Centre of Quantum Computation and Intelligent Systems, UTS. He has published over 30 research papers in top-tier journals and conferences, including the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Cybernetics, *Pattern Recognition*, IJCAI, ICDE, ICDM, SDM, CIKM, and PAKDD. His research interests include data mining and machine learning.