IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Joint Model Feature Regression and Topic Learning for Global Citation Recommendation

Tao Dai[1], Li Zhu[1], Yifan Wang[1], Hongfei Zhang[1], Xiaoyan Cai[2], Yu Zheng[3]

[1]School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[2]School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi, 710072, China
[3]College of Information Engineering, Northwest A&F University, YangLing, Shaanxi, 712100, China

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

This work was partially support by National Natural Science Foundation of China (Project No. 61373046) and Natural Science Basic Research Plan in Shaanxi Province of China (Project No. S2015YFJM2129).

ABSTRACT Citation recommendation has gained increasing attention in recent years. In practice, researchers usually prefer to cite the most topic-relevant articles. Nevertheless, how to model the implicit correlations between topics and citations is still a challenging task. In this paper, we propose a novel citation recommendation model, called TopicCite, which mines such fine-grained correlations. We extract various citation features from citation network, and integrate the learning process of feature regression with topic modeling. At the recommendation stage, we expand the folding-in process by adding the topic influence of papers that correlated with user-provided information. TopicCite can also be considered a technique for extracting topic-related citation features from manually defined citation features, which can essentially improve the granularity of pre-extracted features. In addition, the unsupervised topic model is supervised and mutually reinforced by abundant citation features in TopicCite; thus, the proposed model can also extract more reliable topic distributions from citation data, which brings a new perspective to topic discovery on linked data. The experimental results on the AAN and DBLP datasets demonstrate that our model is competitive with state-of-the-art methods.

INDEX TERMS Citation recommendation, topic model, feature regression

I. INTRODUCTION

Searching suitable references is a time-consuming task for researchers, especially in large and rapidly growing volumes of published scientific databases. For most researchers, the common way to find reference papers is to search for keywords through an online literature search engine, such as Web of Science, Scopus and Google Scholar. The problem with keyword-based searching is that it can return ambiguous results since keywords are too short to represent the relevant papers. Another way to find reference papers is to obtain a small number of papers as starting points and to trace the cited references. However, analyzing all of the cited references to check whether this approach is helpful or not is very labor-intensive because researchers usually need to greatly increase the tracing depth before obtaining the desired papers. Moreover, some important related works could be missed because of page limitation or for other reasons. To solve these problems, citation recommendation that meet various personalized query demands and provide researchers with expected citations, is highly demanded.

According to different query type, citation recommendation can be divided into two main categories: global recommendation [1,2,3,4,5,6,40,41,44] and inline recommendation [6,7,8,9]. The global citation recommendation predicts citations that relevant to a manuscript. It uses an entire part of a manuscript, such as the title, author or venue; thus, it can provide researchers with a panoramic view of the related references. In contrast, inline citation recommendation analyzes the local context of each placeholder to capture its specific information requirements, which could be too short or ambiguous for a query. In addition, selecting the size of each context window is also non-trivial. Therefore, we focus only on global citation recommendation in this work.

Traditionally, whether a paper should be considered as a citation can be measured by the pairwise similarity between a manuscript and the candidate papers. However, solely relying on one type of pairwise similarity is apparently too coarse to serve as a good measurement. A more effective approach is to consider various pairwise similarities as citation features, and applies regression models to learn continuous functions to estimate the relevance of the papers. Many regression-based methods have been proposed in recent years. Strohman *et al*. [1] applied a linear regression method to recommend citations. According to their study, neither of their features performed well in isolation. Livne *et al*. [6] proposed CiteSight, in which the recommendation is achieved by ranking candidate papers through gradient-boosted regression. Nevertheless, there is still some difficulty when applying regression to citation recommendation. First, the common regression models only learn a set of feature weights, which might not capture the characteristics of the citing activity. Researchers usually cite a paper for different reasons in different situations. For example, one researcher cited a paper because it was published in an influential venue, while another researcher cited the paper because it was work performed by a coauthor. It is reasonable that the above two citing activities have different citation patterns, which leads to different feature weights. Second, existing citation features are all defined manually; thus, the types of features are usually limited. How to automatically develop more features is an essential problem for regression-based citation recommendation.

As a high-level feature of text, topics learned by topic models, such as PLSA [10] and LDA [11], can naturally represent the group information of documents; thus, they can reveal the interests of researchers. Many topic models have been proposed and applied in many fields [2,12,13,14]. For citation recommendation, some approaches [2,13,15] take the topic as a high-level feature and use topical similarity to find the most appropriate papers. However, the same as other citation features, topical similarity between papers is also too coarse when applied to citation recommendation.

To address the aforementioned drawbacks of the existing methods, we propose a novel model, named TopicCite, to explore the correlations between the citation features and topics to provide effective citation recommendation. We extract various citation features from bibliographic data and formulate a joint optimization problem with feature regression and topic learning, in such a way that these two modules can be beneficial to each other. The feature weights obtained by the feature regression can assist in finding high quality topics, and high quality topics in turn will result in more accurate weights for measuring the importance of the features. Different from most existing approaches, our model inherits the merits of both discriminative models (feature regression) and generative models (topic learning); thus, it can also extract more reliable topic distributions for citation data. The proposed model is scalable to incorporate any pairwise citation features, and it can also be considered to be a technique to extract fine-grained features from manually defined features, which can essentially improve the granularity of existing features. TopicCite can also be used for other topic-related problems, which brings a new perspective for topic discovery on linked data.

The main contributions of this paper are summarized as follows.

(1) We propose a novel model that jointly combines feature regression with topic models for citation recommendations and topic extraction.

(2) We propose an effective algorithm to solve the joint optimization problem of feature regression and topic learning, which can learn the topic distribution and feature weights simultaneously. We also analyze the time complexity and prove the convergence of the learning algorithm.

(3) Thorough experimental studies on AAN and DBLP datasets are conducted to validate the effectiveness of the proposed model.

The remainder of this paper is organized as follows. Section II presents the related work. Section III introduces the used features and task definitions. Section IV presents our proposed model and methods for parameter learning and recommendation. Section V presents the experimental results. The paper is concluded in Section VI.

II. RELATED WORK

Many citation recommendation methods have been proposed in the literature. These studies can be divided into collaborative filtering (CF)-based methods, graph-based methods and topic similarity-based methods.

A. CF AND CBF BASED CITATION RECOMMENDATION

As one of the most successful recommendation approaches, collaborative filtering is used for citation recommendation. McNee *et al*. [4] first used the collaborative filtering (CF) technique to recommend research papers. Their work is based on a citation web, which is a social network based on citation relationships. They proposed four types of collaborative filtering methods to recommend research papers, including co-citation matching, user-item CF, item-item CF and native Bayesian classifier. Torres *et al*. [16] explored both the social relationships of papers and the content of papers by a hybrid algorithm that combines both CBF and CF. The CBF part considers the text of active papers as input, while the CF part takes the citations from active papers as input. Pohl *et al*. [17] considered downloaded activity as citation activity and recommended users with the most co-accessed papers. They found that co-access provides better coverage than co-citation. Yang *et al*. [18] assumed that users have similar reading interests when they rank common scholarly papers, and they proposed a ranking-oriented collaborative filtering approach. Sugiyama *et al*. [19] constructed user profiles from the paper list that they published and predicted papers by extracting user research preferences. The user profile is enhanced through not only past publications but also papers that cite the work of the user. Li *et al*. [42] proposed a conference paper recommendation method based on CBF. The method extracts various pairwise features and applied pairwise learning to a rank model to predict papers that meet the preferences of the users.

B. GRAPH-BASED CITATION RECOMMENDATION

Open access bibliographic databases, such as AAN and DBLP, are usually composed of paper attributes such as publishing years, authors and venues, which can be considered as complex and heterogeneous networks. More recently, graph models have been applied for citation and document recommendation. Gori and Pucci [20] constructed a homogenous citation graph and applied the PageRank algorithm to recommend scientific papers. Meng *et al*. [5] considered topics as particular nodes and built a four-layer heterogeneous publication graph, and then, they applied a random walk algorithm to recommend papers. Jardine and Teufel [21] extended the bias and transition probabilities of PageRank by considering topic distributions that were extracted from papers to predict scientific papers. Compared with ranking papers by whole link information on graphs, the node similarities on the sub-structures of a document network are much easier to compute, and they can reveal more explicit citation patterns. Sun *et al*. [22] introduced the concept of meta-path, which is a sequence of nodes in a network. They showed that meta-path-based score can obtain achievable performance for similarity search. Ren *et al*. [3] extracted various meta-path-based features from citation graphs and proposed a hybrid model, called ClusCite, which combines nonnegative matrix factorization (NMF) with authority propagation. Guo *et al*. [38] extracted fine-grained co-authorship from citation graphs and recommended papers by graph-based paper ranking in a multi-layered graph. They further expanded the ranking approach with mutually reinforced learning for personalized citation recommendation [39]. Recently, there are emerging graph embedding [47,48] or mining algorithms [49,50,51] for graph analytics, which can be adapted to paper recommendation as well.

C. TOPIC DISCOVERING IN LINKED DATA

As a high-level representation of text, the topic distribution can be considered to be a feature to measure the similarity between a pair of papers. Many researchers have extended topic modeling by integrating link information. Cohn and Hoffman [23] presented a mix topic model, called PHITS, for both text and links. This method embeds terms and links into the same latent topic space, and it can learn the distribution of links/citations and terms simultaneously. Erosheva *et al*. [15] extended LDA and proposed Link-LDA, which follow the same joint learning process from PHITS. Both PHITS and Link-LDA consider links that take the same generative process as terms. Nallapati *et al*. [2] proposed Link-PLSA-LDA to address the document dependency problem that exists in the citation recommendation task. The intuition is that explicit citations can better capture the topic distributions of the documents, while a topic model using link information can improve the performance on hyperlink or citation prediction. Mei *et al*. [24] proposed a topic learning framework called NetSPLSA that introduces graph regularization into PLSA. Chang and Blei [25] proposed RTM, which constrains the topics of the documents with the links between them and suggests citations according to the Hadamard product of the accumulative topic distribution of words between papers in a citation pair.

III. FEATURE EXTRACTION AND TASK DEFINITION

A. FEATURE EXTRACTION

Because of containing rich text information, there are many features that can be extracted from a bibliographic dataset. In CiteSight [6], they extracted many citation features and examined their usefulness. Without loss of generality, we choose some of the same citation features in CiteSight which fit our recommendation scenario. Moreover, we additionally extract meta-path features from heterogeneous bibliographic network because it is easy to obtain abundant citation features by using meta-path. The extracted features are summarized as follows:

Title/Abstract/Keywords similarity: We calculate the TFIDF vectors of the title, abstract and keywords of each paper. We then calculate the similarity between papers in citation pairs using the cosine similarity measure.

Citation count: The citation count measures the importance of a paper. There is no doubt that authors will always prefer important articles to cite. We obtain the citation count of each cited paper from Google Scholar. It should be noted that citation pairs that contain the same cited paper will have the same values by this measurement.

Author similarity: This feature is calculated by the Jaccard index of authors in citation pairs. The main function of this feature is to recommend papers that contain similar authors in a manuscript because authors who are coauthors usually work on the same research field.

Author history: For each cited paper, we calculate its mean cited count of authors in citing papers. The reason for choosing this feature is that the authors will usually cite the same

2169-3536 (c) 2017 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

influential articles in their research fields [26].

Venue relevancy: We obtain the relevancy of venues in a citation pair by calculating the citing frequency from the papers in the venue of the citing paper to the papers in the venue of the cited paper. This feature captures the tendency of authors to cater their results to specific outlets [27].

Meta-path similarity: We extract various meta-path-based features from the dataset. We select 15 different meta-paths, including $PAAP$, $PAVP$, $PVAP$, $(PXP)^y$, $PXP \rightarrow P$, $PXP \leftarrow P$, where $X=\{A,V,T\}$ and $y=\{1,2\}$. We choose both PathSim [22] and a random-walk based measure [28] to calculate the meta-path-based features, since PathSim can only be applied for symmetric meta-paths.

Based on the above feature definition, we can obtain a total of 29 citation features. These valuable features will be used in the regression part of our TopicCite model.

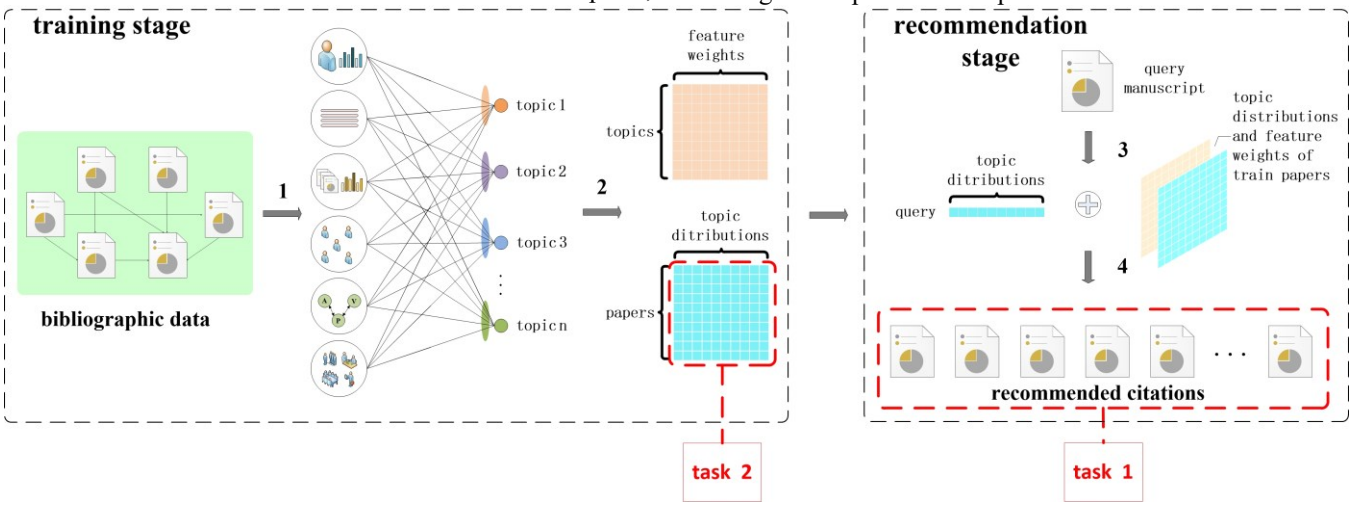


FIGURE 1. **A diagram of the citation recommendation and the topic extraction process in TopicCite**

B. TASK DEFINITIONS

We now formally define the tasks of this paper as follows:

Task 1 (Citation Recommendation): Given a paper collection D, a citation network G and a query manuscript q, which contain words, authors and venues information, the task *Citation Recommendation* is to find most relevant papers for q.

Task 2 (Topic Extraction in Citation Network): Given a paper collection D and a citation network G, the task *Topic Extraction in Citation Network* is aimed at finding K major topics using both the content and link information.

To demonstrate the above two tasks more clearly, we illustrate the computation process of our TopicCite model in Figure 1. It can be seen that there are four essential steps to accomplish the two tasks.

1) Feature extraction. We extract various citation features from bibliographic data according to the features defined in Section III.A.
2) Joint learning feature weights and topic distributions. In this step, feature regression and topic modeling will mutually reinforce to learn the feature weights and topic distributions simultaneously.
3) Learning topic distributions for the query manuscript using folding-in.
4) Recommending citations for the query manuscript. We compute the citation features for the query, and then, we calculate the citation scores for the candidate papers. The papers with higher scores will be recommended as citations.

C. TERMINOLOGY DEFINITION

This subsection defines the notations that are frequently used in this paper. A word or term is a linguistic building element for a paper and is represented by w. A document or paper is a sequence of words and is represented as $d=\{w_1,w_2,...,w_{N_m}\}$, where N_m is the total number of words in a paper. A collection of papers is denoted by $D=\{d_1,d_2,...,d_M\}$, where M is the overall number of papers. C is the set that contains all of the training pairwise papers, including the observed and selected unobserved citation pairs. A query is denoted as $q=[q_w,q_A,q_v]$, where $q_w=\{w_1,w_2,...,w_{N_q}\}$ is a sequence of words, and q_A and q_v are the author and venue information provided by the user, respectively.

TABLE I.
NOTATIONS

Notation	Description
D	Paper collection
M	Number of papers
N_m	Number of words in a paper
$Y_{i,j}$	A training observed or unobserved citation $d_i \rightarrow d_j$
C	A set that contains all citation pairs
K	Number of topics
L	Number of features
θ	The multinomial distribution of topics over papers
φ	The multinomial distribution of words over topics
z_k	The k-th topic
w_e	The e-th word in a paper
$f_{i,j}^l$	The l-th feature for a citation pair $d_i \rightarrow d_j$
$\lambda_{k,l}$	The weight for the l-th feature in topic z_k
q	The query manuscript
q_w^n	The n-th word in q.
q_A	The provided author information in q.
q_V	The provided venue information in q.

As a high-level representation of a paper, the topic usually consists of two types of multinomial distributions: $P(z|d)$ represents the paper-topic distribution, while $P(w|z)$ denotes the topic-word distribution. We assume that the overall number of topics in the paper set is K. Clearly, we have $\sum_k P(z_k|d_i)=1$ and $\sum_e P(w_e|z_k)=1$. For simplicity, we

represent $P(z_k \mid d_j)$ and $P(w_e \mid z_k)$ as $\theta_{k,j}$ and $\varphi_{e,k}$, respectively. Without disambiguation, we feel that it is necessary to clearly define the frequently used notation for quick reference. Table I presents the notation and the corresponding descriptions.

IV. FEATURE REGRESSION WITH TOPIC LEARNING FOR CITATION RECOMMENDATION

A. MODEL OVERVIEW

Given a query $q=[q_w,q_A,q_v]$, it is desirable to recommend papers that have similar topic distributions as the query because these papers are the most relevant studies for the expectation. In section III, we extracted a set of citation features from citation network. In this subsection, we consider a citation score function that considers both the topic relevance and the citation features to decide the possibility of being citing for candidate papers. Suppose that a paper can be divided into K latent topics, then the citation score function between the query q and a candidate paper d_j is defined as follows:

$$s(q,d_j)=\sum_{k=1}^{K}\theta_{k,q}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f^{(l)}(q,d_j) \tag{1}$$

where $\theta_{k,j}$ denotes the probability of topic z_k for paper d_j. Here, $f^{(l)}(q,d_j)$ is the l-th citation feature, which is a simple similarity measurement between q and d_j, and $\lambda_{k,l}$ represents the weight parameter of $f^{(l)}(q,d_j)$ in topic z_k. The meaning of our citation score function is straightforward: $s(q,d_j)$ has a high score only when the following hold: (1) q and d_j both have a high probability in same topics, and (2) the score of the linear combination for $f^{(l)}(q,d_j)$ is also high in the same topics. This arrangement is in contrast to choosing the best representative citations for each query by using the topic similarity approach, similar to in RTM [25] or CTR [29], which do not account for the citation features. Here, $s(q,d_j)$ is also different from a common linear regression approach [1] because the features are divided into topics and each feature has a different weight for each different topic.

Given a training citation network, we use the value 1 to represent an observed citation relationship Y_{ij}, while 0 indicates unobserved examples. The volume of unobserved examples that are chosen randomly is four times larger than the observed examples, since the overall unobserved examples are too large. To simplify the formula, we represent $f^{(l)}(d_i,d_j)$ as $f^{l}_{i,j}$. Then, the loss function according to the mean-square error (MSE) is defined as follows:

$$\begin{aligned} R &= \sum_{d_i,d_j\in C}(Y_{ij}-s(d_i,d_j))^2 \\ &= \sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}\theta_{kj}\sum_{l=1}^{L}\lambda_{k,l}f^{l}_{i,j})^2 \end{aligned} \tag{2}$$

where C is the training set that contains all citation pairs.

It is important to look at the loss function R from the regression perspective. If we transfer the sum order of topics and citation features, then Eq. 2 can be rewritten as follows:

$$R=\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{l=1}^{L}\sum_{k=1}^{K}\lambda_{k,l}\theta_{k,i}\theta_{k,j}f^{l}_{i,j})^2 \tag{3}$$

From the perspective of regression, the above function of R is actually to assign a suitable feature weight $\lambda_{k,l}$ for each of the *topic-related citation features* $\theta_{k,i}\theta_{k,j}f^{l}_{i,j}$. More specifically, the loss function R can extract $K\times L$ new citation features $\theta_{k,i}\theta_{k,j}f^{l}_{i,j}$ from the original L citation features $f^{l}_{i,j}$. Since the topic dimension K (50~300) is usually larger than L (<50), our proposed method can significantly increase the number of available citation features, which consequently improves the granularity of the original citation features.

Our goal is to obtain a minimal error by optimizing the loss function R. Traditionally, we can learn the topic distribution θ by using a topic model first, and then fix θ to derive λ by linear regression. However, such a practice neglects the link information when learning the topics; thus, it cannot precisely measure the contribution of individual features to different topics. To overcome this problem, we integrate feature regression into the topic learning process. Specifically, given a topic model whose objective function is T, we formulate a joint optimization problem, called TopicCite, as follows:

$$\min\,(1-u)\frac{1}{2}R-uT \tag{4}$$

The R in the first part of Eq. 4 is defined in Eq. 2, which measures the benefits of both the topic relevance and the citation features in the citation pairs. The second part T of Eq. 4 measures the topic distribution for individual papers. The parameter u is a bias term. By minimizing Eq. 4, we can simultaneously obtain the optimal parameters for citation recommendation and topic learning. Generally, we can use any topic model, such as PLSA [10] or LDA [11], which are the two most popular statistical topic models. For simplicity, we adopt PLSA in this paper.

B. THE LEARNING PROCESS OF THE JOINT MODEL

In this section, we derive a detailed learning algorithm for the joint optimization problem. By integrating feature regression with PLSA, TopicCite is formulated as follows:

$$\begin{aligned} \min_{\varphi,\lambda,\theta}\ & (1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f^{l}_{i,j})^2 \\ & -u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\log\sum_{k=1}^{K}\varphi_{e,k}\theta_{k,i} \\ & s.t.\ \sum_{k=1}^{K}\theta_{k,i}=1;\ \sum_{e=1}^{N_m}\varphi_{e,k}=1; \end{aligned} \tag{5}$$

where N_m is the number of terms in a paper, and n_{d_i,w_e} is the total number of terms w_e in d_i.

The parameter estimation of the original PLSA uses the expectation maximization (EM) algorithm [30]. In fact, it should be noted that the second part of Eq. 5 is the original likelihood function of PLSA; thus, theoretically, we can still apply the EM algorithm to estimate the parameters in Eq. 5. In E-step, PLSA calculates the conditional probability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

$P(z_k \mid d_i, w_e)$ according to

$$P(z_k \mid d_i, w_e) = \frac{\varphi_{e,k}\theta_{k,i}}{\sum_{k=1}^{K}\varphi_{e,k}\theta_{k,i}} \tag{6}$$

Then, in M-step, Eq. 5 is transformed into Eq. 7 by employing Jensen's inequality, which is shown as follows:

$$\begin{aligned}\min_{\varphi,\lambda,\theta}\ & (1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l})^2 \\ & -u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\sum_{k=1}^{K}P(z_k|d_i,w_e)\log\varphi_{e,k}\theta_{k,i} \\ & s.t.\ \sum_{k=1}^{K}\theta_{k,i}=1;\ \sum_{e=1}^{N_m}\varphi_{e,k}=1;\end{aligned} \tag{7}$$

The original M-step of PLSA takes partial derivatives and leads to closed form solutions for θ and φ. Unfortunately, we do not have a closed form solution for θ in TopicCite. The reason is that the objective function for θ in Eq. 7 is non-convex. However, if we consider each $\theta_{k,i}$ separately, it is convex. In this paper, we develop an alternative algorithm based on gradient descent to calculate each $\theta_{k,i}$.

1) UPDATE RULE FOR φ

First, we fix the paper-topic distribution θ and the feature weights λ to derive the topic-word distribution φ. Then, Eq. 7, which is related to φ, converts into the following optimization problem:

$$\begin{aligned}\max_{\varphi}\ & \sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\sum_{k=1}^{K}P(z_k|d_i,w_e)\log\varphi_{e,k}\theta_{k,i} \\ & s.t.\ \sum_{e=1}^{N_m}\varphi_{e,k}=1;\end{aligned} \tag{8}$$

It can be seen that the objective function related to φ is the same function as in the original PLSA. Therefore, we can directly apply the update rule of each $\varphi_{e,k}$ in PLSA as follows:

$$\varphi_{e,k} = \frac{\sum_{i=1}^{M}n_{d_i,w_e}P(z_k \mid d_i, w_e)}{\sum_{e=1}^{Nm}\sum_{i=1}^{M}n_{d_i,w_e}P(z_k \mid d_i, w_e)} \tag{9}$$

2) UPDATE RULE FOR λ

Next, we fix the topic-word distribution φ and the paper-topic distribution θ to derive the feature weights λ. The optimization problem related to λ is as follows:

$$\min_{\lambda}\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l})^2 \tag{10}$$

By employing the Karush-Kuhn-Tucker (KKT) complementary condition, we can directly obtain the update rule of λ as follows:

$$\lambda_{k,l} = \frac{\sum_{d_i,d_j\in C}Y_{ij}R_0 - \sum_{d_i,d_j\in C}R_0R_1}{\sum_{d_i,d_j\in C}R_0^2} \tag{11}$$

where R_0 and R_1 are defined as follows:

$$R_0 = \theta_{k,i}\theta_{k,j}f_{i,j}^{l} \tag{12}$$

$$R_1 = (\sum_{a=1}^{K}\theta_{a,i}\theta_{a,j}\sum_{x=1}^{L}\lambda_{a,x}f_{i,j}^{x})\Big|_{(a,x)\neq(k,l)} \tag{13}$$

3) UPDATE RULE FOR θ

Finally, we fix the topic-word distribution φ and the feature weights λ to derive the paper-topic distribution θ. By applying the Lagrangian multiplier method, the optimization problem related to θ is as follows:

$$\begin{aligned}\min_{\theta}J(\theta) = \min_{\theta}\ & (1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l})^2 \\ & -u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\sum_{k=1}^{K}P(z_k|d_i,w_e)\log\varphi_{e,k}\theta_{k,i} \\ & +\sum_{i=1}^{M}\tau_i(1-\sum_{k=1}^{K}\theta_{k,i})\end{aligned} \tag{14}$$

where τ_i is the Lagrangian multiplier for each document d_i to penalize the constriction $\sum_{k=1}^{K}\theta_{k,i}=1$. Define C_i as the paper set that is cited by d_i; then, we obtain the gradient of $\theta_{k,i}$ as follows:

$$\begin{aligned}\frac{\partial J(\theta_{k,i})}{\partial\theta_{k,i}} = & -\sum_{d_j\in C_i}(1-u)Y_{ij}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l} \\ & +\sum_{d_j\in C_i}(1-u)\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l}R_2 \\ & -\frac{u\sum_{e=1}^{N_m}n_{d_i,w_e}P(z_k|d_i,w_e)}{\theta_{k,i}}-\tau_i\end{aligned} \tag{15}$$

where R_2 is defined as

$$R_2 = \sum_{a=1}^{K}\theta_{a,i}\theta_{a,j}\sum_{x=1}^{L}\lambda_{a,x}f_{i,j}^{x} \tag{16}$$

We then derive the formula of the Lagrangian multiplier τ_i. By employing the Karush-Kuhn-Tucker (KKT) complementary condition, we obtain the following derivations:

$$\begin{aligned}\tau_i\theta_{k,i} = & -\sum_{d_j\in C}(1-u)Y_{ij}\theta_{k,i}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l} \\ & +\sum_{d_j\in C}(1-u)\theta_{k,i}\theta_{k,j}\sum_{l=1}^{L}\lambda_{k,l}f_{i,j}^{l}R_2 \\ & -u\sum_{e=1}^{N_m}n_{d_i,w_e}P(z_k|d_i,w_e)\end{aligned} \tag{17}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

Since $\sum_{k=1}^{K}\theta_{k,i}=1$, the formula of the Lagrangian multiplier τ_i is derived as follows:

$$\begin{aligned}\tau_i &= \tau_i\sum_{a=1}^{K}\theta_{a,i}=\sum_{a=1}^{K}\tau_i\theta_{a,i}\\ &=-(1-u)\sum_{d_j\in C}Y_{ij}R_2+(1-u)\sum_{d_j\in C}R_2^2\\ &\quad -u\sum_{a=1}^{K}\sum_{e=1}^{N_m}n_{d_i,w_e}P(z_a|d_i,w_e)\end{aligned} \tag{18}$$

Replacing τ_i in Eq. 15 with Eq. 18, we then obtain

$$\begin{aligned}\frac{\partial J(\theta_{k,i})}{\partial\theta_{k,i}} &= (1-u)\sum_{d_j\in C_i}(R_3-R_2)(R_2-Y_{ij})\\ &\quad -\frac{u\sum_{e=1}^{N_m}n(d_i,w_e)P(z_k|d_i,w_e)}{\theta_{k,i}}\\ &\quad +u\sum_{a=1}^{K}\sum_{e=1}^{N_m}n_{d_i,w_e}P(z_a|d_i,w_e)\end{aligned} \tag{19}$$

where R_3 is defined as

$$R_3=\theta_{k,j}\sum_{x=1}^{L}\lambda_{k,x}f_{i,j}^{x} \tag{20}$$

According to the gradient descent method, the update rule of $\theta_{k,i}$ is as follows:

$$\theta_{k,i}=\theta_{k,i}+\beta\frac{\partial J(\theta_{k,i})}{\partial\theta_{k,i}} \tag{21}$$

where β is the step size. Next, let us define β as follows:

$$\beta=-\frac{\theta_{k,i}}{(1-u)\sum_{d_j\in C_i}(R_3-R_2)(R_2-Y_{ij})} \tag{22}$$

Then, the final update rule for $\theta_{k,i}$ is derived as follows:

$$\theta_{k,i}=\frac{u\sum_{e=1}^{N_m}n_{d_i,w_e}P(z_k|d_i,w_e)-u\theta_{k,i}\sum_{a=1}^{K}\sum_{e=1}^{N_m}n_{d_i,w_e}P(z_a|d_i,w_e)}{(1-u)\sum_{d_j\in C_i}(R_3-R_2)(R_2-Y_{ij})} \tag{23}$$

Because $\theta_{k,i}$ and $\theta_{k,j}$ are symmetric in Eq. 14, the update rule for $\theta_{k,j}$ is derived as follows:

$$\theta_{k,j}=\frac{u\sum_{e=1}^{N_m}n_{d_j,w_e}P(z_k|d_j,w_e)-u\theta_{k,j}\sum_{a=1}^{K}\sum_{e=1}^{N_m}n_{d_j,w_e}P(z_a|d_j,w_e)}{(1-u)\sum_{d_i\in C_j}(R_4-R_2)(R_2-Y_{ij})} \tag{24}$$

where C_j is the paper set that cites d_j, and R_4 is defined as follows:

$$R_4=\theta_{k,i}\sum_{x=1}^{L}\lambda_{k,x}f_{i,j}^{x} \tag{25}$$

4) LEARNING ALGORITHM OF TOPICCITE

The above updating rules are applied at the M-step of each iteration, until Eq. 7 converges to a local maximum. By setting the starting point, the whole learning process can be repeated. Then, TopicCite takes the next iteration of E-step and M-step, until the whole algorithm converges. When the algorithm stops, the optimal feature weights λ and the topic distributions θ for each citation pair can be learned simultaneously. We summarize the parameter learning of TopicCite in Algorithm 1.

Algorithm 1 Parameter Learning of TopicCite

Input: paper collection D, citation matrix Y, L citation features $\{f^l\}$, topic number K, bias term u.

Output: paper-topic distribution θ, feature weights λ.

Initialize: θ, φ, λ with positive values.

1: **repeat**
2: **E-step**
3: Calculate the local expectation of the topics by Eq. 6;
4: **end E-step**
5: **M-step**
6: Update the topic-word distribution φ by Eq. 9;
7: Update the feature weights λ by Eq. 11;
8: **repeat**
9: Update the paper-topic distribution θ for each citation pair by Eq. 23 and Eq. 24;
10: **until** objective in Eq. 7 converges
11: **end M-step**
12: **until** objective in Eq. 5 converges

C. CITATION RECOMMENDATION USING TOPICCITE

After the parameter learning, we use Eq. 1 to recommend the citation for a query. However, the implicit query-topic distribution is still unknown. To extract the topic from a new document, many folding-in based methods have been proposed for PLSA [10,12,31]. The deficiency for both of them is that the user preference in the query is neglected. When searching for citations, people can provide some related information such as authors and venues, and we believe that researchers have different expectations when they provide such personalized information. We extend the PLSA folding-in by integrating the topic distribution of the papers that correlated with user-provided authors and venues.

Following the original PLSA, the objective function for a query manuscript q is as follows:

$$\begin{aligned}&\max_{\varphi,\theta}\sum_{n=1}^{N_m}n_{q,q_w^n}\log\sum_{k=1}^{K}\varphi_{q_w^n,k}\theta_{k,q}\\ &s.t.\ \sum_{k=1}^{K}\theta_{k,q}=1;\ \sum_{n=1}^{N_m}\varphi_{q_w^n,k}=1;\end{aligned} \tag{26}$$

where q_w^n is the n-th term in q_w, and n_{q,q_w^n} is the total number of terms q_w^n in q. The folding-in of PLSA is based on an incremental variant of the EM algorithm [10] The basic idea is to fix φ and initialize $P(z\,|\,q)$ randomly then, to take the EM algorithm according to Eq. 27 and Eq. 28.

$$P(z_k\,|\,q,q_w^n)=\frac{\varphi_{q_w^n,k}\theta_{k,q}}{\sum_{k=1}^{K}\varphi_{q_w^n,k}\theta_{k,q}} \tag{27}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

$$\theta_{k,q}=\frac{\sum_{n=1}^{N_m} n_{q,q_w^n} P(z_k \mid q, q_w^n)}{\sum_{k=1}^{K}\sum_{n=1}^{N_m} n_{q,q_w^n} P(z_k \mid q, q_w^n)} \tag{28}$$

When the objective function for the query converges, we calculate $\theta_{k,q}$ as follows:

$$\theta_{k,q}=(1-\alpha)\theta_{k,q_w}+\alpha\frac{1}{N_c}\sum_{d\in D^P}\theta_{k,d} \tag{29}$$

where $D^P = D^A \bigcup D^V$ is the paper set correlated with the user-provided authors and venues. N_c is the total number of documents in D^P. Here, α is a coefficient that evaluates the importance with respect to the user's personalized information. After extracting the query-topic distribution, we calculate L citation features for the query and recommend the papers with the higher scores for the query according to Eq. 1. The citation recommendation process of TopicCite is specified in Algorithm 2.

Algorithm 2 Citation recommendation using TopicCite

Input: paper set D, paper-topic distribution θ of D, topic-word distribution φ, feature weights λ, query $q=[q_w, q_A, q_v]$, total number of topics K, bias term α.
Output: paper set D_r as citations.
Initialize: θ of the query with positive values.
1: **repeat**
2: **E-step**
3: Calculate local expectation of topics for q by Eq. 27;
4: **end E-step**
5: **M-step**
6: Update θ_{q_w} by Eq. 28;
7: **end M-step**
8: **until** Eq. 26 converges
9: Calculate user personalized θ_q by Eq. 29;
10: Calculate L citation features $f^{(l)}(q,d)$ between the query and candidate papers;
11: Calculate citation scores $s(q,d)$ by Eq. 1;
12: Recommend top papers D_r for q as citations by citation scores;

D. TIME COMPLEXITY ANALYSIS

For parameter learning stage, the time complexity is determined by a double loop. In the outer loop, computing the local expectation of the topics by Eq. 6 takes $O((2K-1)\times N_m \times M)$ times. Updating the topic-word distribution φ by Eq. 9 takes $O(K\times(N_m+2M-1))$ times. Updating the feature weights λ by Eq. 11 takes $O(|C|+|C|\times K\times L)$ times. Let us denote $|C^{\Delta}|$ as the overall number of cited papers. Then in the inner loop, updating the paper-topic distribution θ for all citation pairs by Eq. 23 and 24 takes $O(|C|\times K\times N_m+|C|\times(K+L)+|C^{\Delta}|\times K\times L)$ times. Because K and L is much smaller than N_m, M and $|C|$, the time complexity of parameter learning in TopicCite is approximate to $O(n^2)$ for each iteration. For recommendation stage, calculating θ for a query only takes $O(K\times N_m+K)$ times for each iteration, which is nearly $O(n)$. It should be noted that the recommendation in TopicCite is time insensitive because the folding-in process can converge in only a few iterations and all citation features can be easily computed.

E. CONVERGENCE ANALYSIS

In this section, we analysis the convergence of the learning stage. We give the following theorem.

Throrem 1. *The iterations of algorithm 1 lead the objective function of Eq. 5 converges to a local minimum.*

Proof. At the beginning of each t-th iteration, the overall loss error E_1^t can be calculated as:

$$\begin{aligned}E_1^t=&(1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}^{t-1}\theta_{k,j}^{t-1}\sum_{l=1}^{L}\lambda_{k,l}^{t-1}f_{i,j}^{l})^2\\&-u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\log\sum_{k=1}^{K}\varphi_{e,k}^{t-1}\theta_{k,i}^{t-1}\end{aligned} \tag{30}$$

In the step 2~4, algorithm 1 computes the local expectations of topics. According to Jensen's inequality, the loss error E_2^t generated by these step is equal to E_1^t.

$$\begin{aligned}E_2^t=&(1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}^{t-1}\theta_{k,j}^{t-1}\sum_{l=1}^{L}\lambda_{k,l}^{t-1}f_{i,j}^{l})^2\\&-u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\sum_{k=1}^{K}P(z_k^t|d_i,w_e)\log\varphi_{e,k}^{t-1}\theta_{k,i}^{t-1}\\=&E_1^t\end{aligned} \tag{31}$$

In the step 6 and step 7, algorithm 1 directly computes the t-th φ and λ according to the saddle point of gradient. Therefore, the loss error E_3^t generated by these step is less than E_2^t.

$$\begin{aligned}E_3^t=&(1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}^{t-1}\theta_{k,j}^{t-1}\sum_{l=1}^{L}\lambda_{k,l}^{t}f_{i,j}^{l})^2\\&-u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\sum_{k=1}^{K}P(z_k^t|d_i,w_e)\log\varphi_{e,k}^{t}\theta_{k,i}^{t-1}\\\le&E_2^t\end{aligned} \tag{32}$$

In the step 8~10, algorithm 1 calculates the t-th θ according to gradient descent. The generated loss error E_4^t is less than E_3^t.

$$\begin{aligned}E_4^t=&(1-u)\frac{1}{2}\sum_{d_i,d_j\in C}(Y_{ij}-\sum_{k=1}^{K}\theta_{k,i}^{t}\theta_{k,j}^{t}\sum_{l=1}^{L}\lambda_{k,l}^{t}f_{i,j}^{l})^2\\&-u\sum_{i=1}^{M}\sum_{e=1}^{N_m}n_{d_i,w_e}\sum_{k=1}^{K}P(z_k^t|d_i,w_e)\log\varphi_{e,k}^{t}\theta_{k,i}^{t}\\\le&E_3^t\end{aligned} \tag{33}$$

According to Jensen's inequality, we can also obtain $E_4^t \ge E_1^{t+1}$. In summary, the loss errors in Eq. 5 are maintained as descending during the iterations of algorithm 1 as:

$$E_1^1=E_2^1\ge E_3^1\ge E_4^1\ge E_1^2=\ldots\ge E_1^t=E_2^t\ge E_3^t\ge E_4^t\ge E_1^{t+1}=\ldots \tag{34}$$

From the above analysis, we can see that the overall loss error of Eq. 5 decreases monotonically. Therefore, algorithm 1

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

will lead the objective function of Eq. 5 converges to a local minimum.

V. EXPERIMENTS AND RESULTS

A. DATASETS

To evaluate our proposed model, we choose two bibliographic datasets, AAN and DBLP, which have different sizes of research publications in different research fields.

· AAN dataset[1]: Radev *et al*. [32] established the ACL Anthology Network (AAN) dataset, which contains full text information of conference and journal papers in the computational linguistics and natural language processing field. We use a subset of a 2012 release that contains 13,885 papers published from 1965 to 2012. For evaluation purposes, we divide the entire dataset into two disjoint sets, where papers published before 2012 are regarded as the training set (12,762 papers) and the remaining papers are placed in the testing set (1,123 papers).

· DBLP dataset[2]: DBLP is a well-known online digital library that contains a collection of bibliographic entries for articles and books in the field of computer science and related disciplines. We use a citation dataset that was extracted and released by Tang *et al*. [33]. Instead of employing a full dataset, we choose a subset since some samples miss complete references. The papers published before 2009 are considered as the training set (29,193 papers), and the papers published from 2009 to 2011 are considered as the testing set (2,869 papers).

Each paper in the dataset is pre-processed by removing stop words, and stemming is performed using Porter stemmer. To reduce the impact of short words, we removed the words that consist of less than 2 characters and appear less than ten times in the datasets. Terms and key phrases are extracted by computing the TF-IDF score. We then reduce the dimension of the TF-IDF matrix into 5,000 dimensions according to a fast PCA algorithm [34]. Table II summarizes the statistics of these two datasets.

TABLE II.
STATISTICS OF AAN AND DBLP

		Papers	Authors	Venues	Citations
AAN	Train	12,762	9,799	467	68,475
	Test	1,123	1,557	32	10,437
DBLP	Train	29,193	32,541	751	397,316
	Test	2,869	3,184	104	31,501

B. EVALUATION METRICS

To evaluate the quality of the recommendations, we use the citation information of the training papers to train our model, and the reference lists of the testing papers are used as the ground truth. Following common practice, we employ the following evaluation metrics:

· Precision and Recall, which are two commonly used metrics for information retrieval field. Precision@N (P@N) measures the percentage of the retrieved citations that is relevant to the ground truth in the top-N recommendation list. Recall@N (R@N) measures the rate of the real citations that are retrieved in the top-N recommendation list. These two metrics are calculated as follows:

$$Precision = \frac{\sum_{d \in Q(D)} |R(d) \cap T(d)|}{\sum_{d \in Q(D)} |R(d)|} \tag{35}$$

$$Recall = \frac{\sum_{d \in Q(D)} |R(d) \cap T(d)|}{\sum_{d \in Q(D)} |T(d)|} \tag{36}$$

where D is a paper set, and $Q(D)$ is the set of test citing papers. $T(d)$ is the ground truth citations that are contained in paper d, and $R(d)$ is the citation recommended for the test paper d.

· Mean Reciprocal Rank (MRR) is the average of the reciprocal ranks of the results for the test paper set $Q(D)$. The reciprocal rank of a result is the multiplicative inverse of the rank of first matched recommended citations. MRR is computed as follows:

$$MRR = \frac{1}{|Q(D)|} \sum_{d \in Q(D)} \frac{1}{rank_d} \tag{37}$$

where $rank_d$ is the position of the first correct result of test paper $d \in Q(D)$.

MRR accounts for the rank of the recommended citation, and consequentially, it heavily penalizes the retrieval results when the relevant citations are returned at low rank.

C. COMPARISON WITH OTHER APPROACHES

The compared methods are summarized as follows. The provided parameters are the optimal value tuned by our experiments.

· BM25 [35]: BM25 is a well-known ranking method for measuring the relevance of matching documents to a query based on the text. We calculate the text similarity between the papers by using both TF and IDF for BM25. According to our experimental results, we set bias term b to 0.76. The bias term k_1 and k_2 are set to 1.3 and 650, respectively.

· PopRank [36]: PopRank is an extension of PageRank, which adds a popularity propagation factor (PPF) to each citation to an object, and utilizes the author-paper relationship and the publication venue-paper relationship to rank the authoritative papers. We set the PPF factor to 0.3 and the threshold to 0.01 in PopRank.

· Random walks (RW) [5]: RW is also a PageRank based method that is conducted on a heterogeneous graph that consists of papers, authors, terms and topics. The topics in RW are extracted by the LDA model. The restarting probability is set to 0.8, and the topic number is set to 70 in RW.

· PWFC [38]: PWFC exploits fine-grained co-authorships among authors, and builds a three-layer graph to perform random walk based ranking. The optimal parameters are set same as in [38].

· RankSVM [37]: RankSVM assumes that all observed citation relations are positive examples and unobserved ones are negative, and it conducts pairwise classification to recommend citations. We minimize the leave-one-out error in RankSVM on the training set by setting the trade-off to 0.005.

· ClusCite [3]: ClusCite assumes that citation features should be organized into different groups, and each group contains its own behavior pattern to represent the research interest. This method combines NMF and network regularization, to learn group and authority information for citation recommendation.

[1] clair.eecs.umich.edu/aan/
[2] www.aminer.cn/

To ensure fairness in the comparison, we use all extracted citation features in this paper for ClusCite. The number of interest groups is set to 80. The regularization factors c_p and c_w are set to 10^{-6} and 10^{-7}, respectively. We also set both authority factors λ_A and λ_V to 0.3.

· ConceptPRec [45]: ConceptPRec uses Paragraph Vector [46] to learn deep representations of papers. The recommendation scores are computed by cosine similarity between the query manuscripts and candidate papers. The optimal parameters are set same as in [45].

· TopicSim: We use the original PLSA to derive the topic information; then, we recommend papers that have high topic relevance with the query. The optimal topic number is set to 75.

· RTM [25]: RTM is an extension of sLDA [43], which uses links as supervision to train the LDA model; then, it uses the sigmoid function with the Hadamard product of topic distributions to recommend papers. The topic number is set to 70.

· Linear regression (LR) [6]: We discard the topic information in TopicCite and apply only the feature regression method to recommend citations, which is the same approach as in CiteSight.

TABLE III.
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON AAN

Method	MRR	P@10	P@20	P@25	R@25	R@50	R@75	R@100
BM25	0.3356	0.1680	0.1053	0.0889	0.1613	0.3078	0.3494	0.3702
PopRank	0.1799	0.0828	0.0558	0.0130	0.1564	0.2539	0.3172	0.3331
RW	0.4463	0.1731	0.1348	0.0937	0.2071	0.3174	0.3552	0.4053
PWFC	0.4585	0.2059	0.1559	0.1268	0.2284	0.3162	0.3795	0.4246
RankSVM	0.4697	0.2392	0.1839	0.1523	0.2394	0.3495	0.3978	0.4520
ClusCite	0.5619	**0.2509**	0.1932	0.1632	**0.2724**	0.3671	0.3953	0.4702
ConceptPRec	0.1784	0.1029	0.0652	0.0325	0.1242	0.2153	0.2389	0.2492
TopicSim	0.1477	0.0799	0.0603	0.0250	0.1182	0.2042	0.2127	0.2264
RTM	0.4358	0.1784	0.1247	0.0919	0.1892	0.2942	0.3654	0.3991
LR	0.4203	0.1306	0.0992	0.0464	0.1358	0.2321	0.3250	0.3489
TopicCite	**0.5713**	0.2497	**0.2084**	**0.1795**	0.2673	**0.3749**	**0.4297**	**0.5035**

TABLE IV.
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON DBLP

Method	MRR	P@10	P@20	P@25	R@25	R@50	R@75	R@100
BM25	0.3492	0.1741	0.1259	0.0992	0.1806	0.3281	0.3528	0.3892
PopRank	0.1951	0.0953	0.0742	0.0398	0.1789	0.2692	0.3221	0.3598
RW	0.4572	0.1838	0.1431	0.1057	0.2182	0.3298	0.3703	0.4273
PWFC	0.4614	0.2137	0.1693	0.1396	0.2382	0.3349	0.4062	0.4390
RankSVM	0.4781	0.2254	0.1791	0.1476	0.2431	0.3572	0.4186	0.4682
ClusCite	0.5763	0.2617	0.2149	0.1682	**0.3027**	**0.4317**	0.4482	0.5075
ConceptPRec	0.1694	0.1122	0.0973	0.0484	0.1315	0.2254	0.2413	0.2541
TopicSim	0.1581	0.0914	0.0791	0.0196	0.1289	0.2193	0.2210	0.2351
RTM	0.4481	0.1745	0.1309	0.0934	0.2003	0.3154	0.3689	0.4198
LR	0.4359	0.1485	0.1098	0.0681	0.1591	0.2557	0.3482	0.3732
TopicCite	**0.5847**	**0.2674**	**0.2292**	**0.1863**	0.2894	0.4263	**0.4628**	**0.5273**

The experiments were carried on a workstation with a 3.40 GHz Intel® Core™ i7-6700 quad-core processor, 32 GB of RAM. The results are shown in Tables III and IV. It can be observed that for the trend in the precision of these methods, all of them decline, while the trend of the recall continues to increase when the size of the recommended list increases. It is evident that our TopicCite significantly outperforms other approaches, except for only ClusCite in some cases. For example, compared with ClusCite, on average, TopicCite improves the accuracy by 1.67%, 7.87% and 2.13% relative to MRR, P@20 and R@50, respectively, on the AAN dataset. The improvements show that TopicCite is a promising recommender approach.

There are two connections between ClusCite and our method. One is they both use linear combination of features. The other is they both divide the linear combination into K latent dimensions. Although the above two connections exist, our method still different from ClusCite due to the following reasons. Firstly, ClusCite only considers the K latent groups of citing papers, while our method explores K latent topics for both citing and cited papers. Moreover, the citation score in our method is restricted by topic relevance between citing and cited papers, which is more reasonable than ClusCite for researchers usually prefer topic related papers to cite. Secondly, the latent groups in ClusCite is learnt though non-negative matrix factorization (NMF), and only latent information of papers is learnt in ClusCite. In contrast, our method explores the latent topics by topic learning, which learns latent information both for papers and terms. Therefore, our method can fully excavate the semantic information in provided text to generate more suitable latent topics. Moreover, topic modeling has more solid statistical foundation comparing with NMF. Thirdly, ClusCite combines authority propagation based on network regularization. As new papers are added, ClusCite needs to recalculate authority scores though whole citation graph, which is a heavy time consumed process. In contrast, our method only need to calculate topic distribution of new papers using folding-in, which is more lightweight than ClusCite. Finally, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

original ClusCite only uses meta-path based features, while our work adds more features into model learning. The results show that these additional features are benefit for improving performance. For example, P@20 in DBLP of original ClusCite that only uses meta-path based feature is 0.1958 [3], while P@20 in DBLP of ClusCite that uses more feature is 0.2149.

RankSVM is the second comparable method to ours. However, RankSVM relies on only manually defined citation features, while our method extracts more fine-grained topic-related citation features among the manually defined citation features. TopicCite shows better performance than RTM, which can be mainly credited to TopicCite considering not only the link information but also various valuable citation features, which are ignored in RTM.

We also observe that BM25 constantly outperforms TopicSim in all cases, which shows that text-based features are more important than topic-based features in finding relevant papers. Although ConceptPRec learns deep representation for papers, its performance is just a little better than TopicSim. The reason is that ConceptPRec is lack of the supervision of link information during training stage. RW shows a clear performance gain over PopRank because RW constructs a more sophisticated graph by extracting latent topics. However, the topics in RM are extracted by the original LDA model, which fully neglects the citation information; thus, it could be biased with respect to the actual topic distribution in the linked datasets. PWFC performs better than RM, for it extracts fine-grained co-author ship to build a multi-layer graph. Comparing with word level similarity in RW, author level similarity in PWFC can better reflect patterns of citing activity.

D. TRAINING TIME ANALYSIS

TABLE V.
TRAINING TIME BETWEEN DIFFERENT METHODS

Method	AAN	DBLP
RankSVM	27,348 s	162,081 s
ClusCite	19,224 s	112,395 s
TopicSim	10,952 s	59,714 s
RTM	15,492 s	90,548 s
LR	6,471 s	37,813 s
TopicCite	12,589 s	73,542 s

In this subsection, we evaluate the training time of TopicCite against RankSVM, ClusCite, TopicSim, RTM and LR. Table V shows the training time (in seconds) of these methods. It can be seen that LR is the fastest method, but its performance is unsatisfied. During topic based methods, our method just spends a little more time than original PLSA (TopicSim). RankSVM and ClusCite are the two most time-consuming methods. Our method is about more than 0.52 and 1.17 times faster to train than ClusCite and RankSVM, respectively.

E. PARAMETER TUNING

In this section, we study the impact of three hyper parameters: u , K and α in TopicCite. Parameter u is a ratio that constitutes a tradeoff between the citation relevance and the topic model; parameter K controls the expected number of topics; and parameter α is a bias with respect to the influence of the topics for the user-provided authors and venues. We provide the parameter tuning of R@100 in the AAN dataset, and other metrics generate similar results in our experiments.

To explore the performance of these hyper parameters, we first empirically fix $u = 0.3$ and $\alpha = 0.2$ to evaluate the effects of varying K . Figure 2 shows the R@100 measured as a function of K . It shows that the precision score becomes higher for larger values of K . This trend occurs because a larger number of topics can better capture the topical relevance for citation pairs. In our experiments, we chose $K = 70$ because a larger K does not appear to give much better results. It should be noted that TopicCite still performs better than TopicSim (PLSA) and LR even when $K = 1$, which demonstrates that *topic-related citation features* $\theta_{k,i}\theta_{k,j}f^{l}_{i,j}$ can significantly improve the recommendation performance.

We then fixed $K = 70$ and $\alpha = 0.2$ to evaluate the effects of varying u . Figure 3 shows R@100 measured as a function of u . We observe that the precision continually increases until u reaches 0.4. Because the optimal bias term for feature regression is larger than the one for topic learning, it can be concluded that feature regression plays a more important role in our model comparing with topic learning.

Finally, we fix $K = 70$ and $u = 0.4$ to analyze the sensitivity of α . As shown in Figure 4, we observe that performance increases until α reaches 0.2. The result indicates that although topics obtained from the historical publications of author will improve recommendation performance, its importance is obvious less than the query manuscript. We analyzed the recommendation results carefully, and found that the reasons can be ascribed into two folds: 1) researchers are not always focus on one research area. As time passed, the topic related to researcher will drift, which reduces the correlation between current research and historical publications of author. 2) some junior researchers only published one scientific, thus there are no topics that generated by historical publications for these authors.

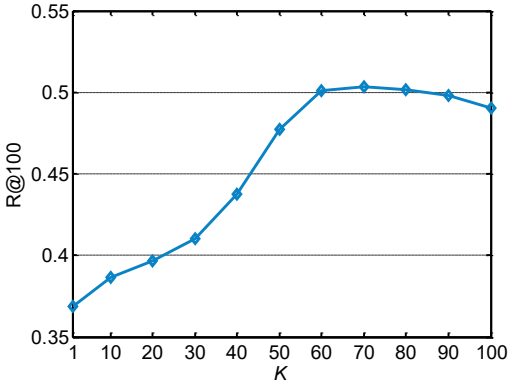


FIGURE 2. The performance impact of *K*

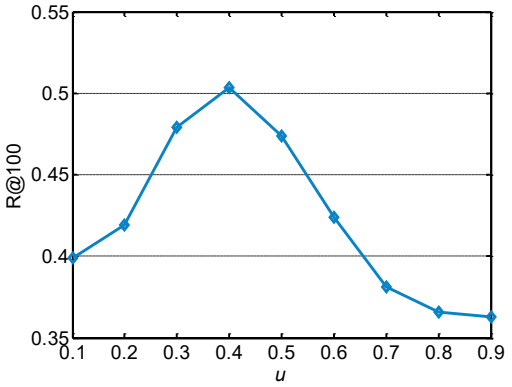


FIGURE 3. The performance impact of u

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access

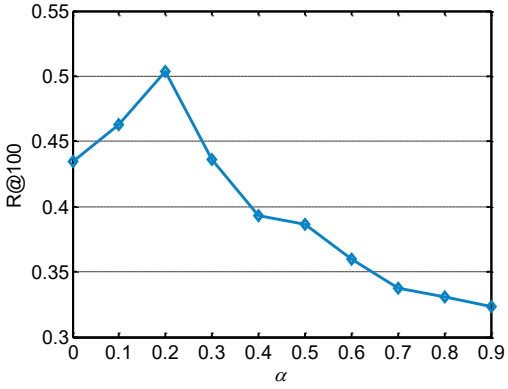


FIGURE 4. The performance impact of α

F. TOPIC ANALYSIS

To have a better understanding of why our framework outperforms other topic models and to validate the goal defined in task 2, we chose a citation pair, W02-1405→A00-1009, from the AAN network and compared the paper-topic distribution extracted by TopicCite, PLSA and RTM. The titles of the two papers are listed in Table VI.

TABLE VI.
A CITATION PAIR IN AAN

	Paper id	Title
Citing	W02-1405	Improving a General-purpose Statistical Translation Engine by Terminological Lexicons
Cited	A00-1009	A Framework for MT and Multilingual NLG Systems based on Uniform Lexico-Structural

Figures 5 and 6 show the paper-topic distribution extracted for the citation pair. We can see that there are distinctively different paper-topic distributions between these methods. TopicCite can extract a more similar paper-topic distribution for a pair of citations. For example, the highly ranked topics for paper W02-1405 are 40, 9, 45 and 91 in Figure 5(a), which are also highly ranked in paper A00-1009 in Figure 6(a). TopicCite uses both link information and various citation features. By considering this information as a regularization term for topic model, TopicCite will force citation pairs to have similar paper-topic distributions, which improves the performance for the citation recommendation. Although RTM can also extract a similar paper-topic distribution for a citation pair, such as topic 45 and 8 for W02-1405→A00-1009 in Figure 5(c) and 6(c), it still has less similar topics compared with TopicCite. As seen in Figure 5(b) and 6(b), PLSA extracts completely different paper-topic distributions for the two papers because the link information is completely ignored. The results show that TopicCite can better reveal the citing-cited relation for a pair of citations from the perspective of the topic distributions.

To reveal the latent topics and to make a comparison with other topic models, we extracted topics and their related words by applying TopicCite, PLSA and RTM on the AAN dataset. Since the AAN dataset is a mixture dataset that contains various research areas such as ontologies, machine translation and semantic representations of spoken language, it is interesting to see whether these hidden topics could reasonably reveal this mixture character. We chose 3 topics from results, and we show the top ranked words in Table VII. It can be seen from the first topic that all of the extracted words are basically the same and can well describe the corresponding topic. For the second topic, different topic models selected considerably different words. For example, words such as "speech, enhance, band" inferred by TopicCite are obviously more meaningful than "signal, speech, mask" inferred by PLSA and "speech, criterion, classical" inferred by RTM. For topic 3, the words extracted by PLSA appear to be too ambiguous to represent a topic, but TopicCite and RTM derive more reasonable words to represent the topic, such as "named entity extract". The results demonstrate that TopicCite can select more meaningful words for each topic than PLSA and RTM, and show that TopicCite has better performance in extracting topics.

VI. CONCLUSIONS

In this paper, we investigate citation recommendation with both text content and citation features. We extracted various citation features from a citation network; then, we proposed a joint feature regression and topic learning model. Based on the feature regression part, our framework can learn more fine-grain feature weights with topic learning to accurately measure the importance of each individual citation feature, and the citation features further help to extract better topic distributions. The process runs in an iterative way for maximum benefit. We then extended the folding-in process to integrate the topic influence of the papers that were related to the user-provided authors and venues to recommend citations. The experimental results validated the effectiveness of our algorithm in both citation recommendation and topic discovery.

In the future, there are many potential directions for this work. There are still more effective features, such as the locations of words and the timing of the papers. We will incorporate these features to improve the performance of our model. In addition, our model does not consider the a priori information of the topic model. We plan to extend our model to handle data with a priori information.

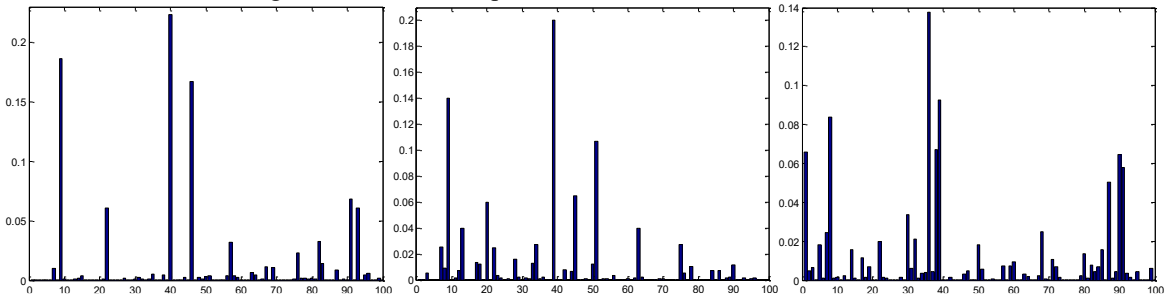
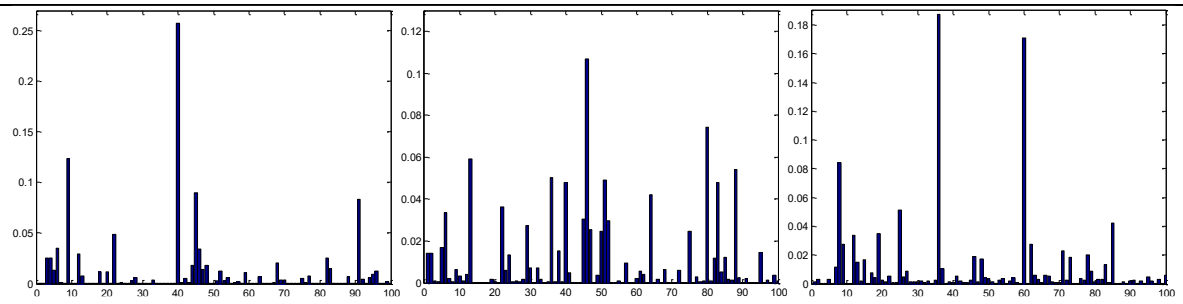


FIGURE 5. Comparison of paper-topic distributions for W02-1405 on AAN

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2884981, IEEE Access



(a) TopicCite (b) PLSA (c) RTM

FIGURE 6. Comparison of paper-topic distributions for A00-1009 on AAN

TABLE VII.
THE REPRESENTATIVE WORDS GENERATED BY TOPICCITE, PLSA AND RTM MODELS FOR AAN

TopicCite			PLSA			RTM		
Topic 1	Topic 2	Topic 3	Topic 1	Topic 2	Topic 3	Topic 1	Topic 2	Topic 3
social	speech	named	social	signal	cluster	social	speech	named
identify	enhance	location	opinions	speech	relation	friends	criterion	relation
discuss	band	entity	annotate	mask	entity	mining	classical	entity
group	filter	extract	mining	weight	tag	group	reduce	text
mining	gain	node	system	power	sentence	annotate	function	link
online	estimate	relation	classify	convert	complete	system	audience	task
opinions	noisy	task	identify	frame	domain	opinions	filter	domain
annotate	quality	find	friends	analysis	label	specific	spectrum	extract
system	signal	mark	group	critical	describe	online	noise	concept
friends	human	domain	discuss	reduce	specific	discuss	mask	model

2169-3536 (c) 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

REFERENCES

[1] T. Strohman, WB. Croft, and D. Jensen, "Recommending citations for academic papers," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2007, pp. 705-706.
[2] RM. Nallapati *et al.*, "Joint latent topic models for text and citations," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 542-550.
[3] X. Ren *et al.*, "Cluscite: Effective citation recommendation by information network-based clustering," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2014, pp. 821-830.
[4] SM. McNee *et al.*, "On the recommending of citations for research papers." In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, ACM, 2002, pp. 116-125.
[5] F. Meng *et al.*, "A unified graph model for personalized query-oriented reference paper recommendation," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, ACM, 2013, pp. 1509-1512.
[6] A. Livne *et al.*, "CiteSight: Supporting contextual citation recommendation using differential search," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, ACM, 2014, pp. 807-816.
[7] Q. He *et al.*, "Context-aware citation recommendation," in *Proceedings of the 19th international conference on World Wide Web*, ACM, 2010, pp. 421-430.
[8] Q. He *et al.*, "Citation recommendation without author supervision," in *Proceedings of the 4th ACM international conference on Web search and data mining*, ACM, 2011, pp. 755-764.
[9] W. Huang *et al.*, "Recommending citations: translating papers into references," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, ACM, 2012, pp. 1910-1914.
[10] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177-196, 2001.
[11] DM. Blei, AY. Ng, and MI. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.
[12] TC. Chou and MC. Chen, "Using incremental PLSI for threshold-resilient online event analysis," *IEEE transactions on Knowledge and Data Engineering*, vol. 20, no.3, pp. 289-299, 2008.
[13] H. Xia *et al.*, "Plink-LDA: Using link as prior information in topic modeling," in *International Conference on Database Systems for Advanced Applications*, Springer, Berlin, Heidelberg, 2012, pp. 213-227.
[14] MA. Haidar and D. O'Shaughnessy, "Unsupervised language model adaptation using LDA-based mixture models and latent semantic marginals," *Computer Speech & Language*, vol. 29, no.1, pp. 20-31, 2015.
[15] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed-membership models of scientific publications," in *Proceedings of the National Academy of Sciences*, vol. 101, suppl.1, pp. 5220-5227, 2004.
[16] R. Torres *et al.*, "Enhancing digital libraries with TechLens+," in *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, ACM, 2004, pp. 228-236.
[17] S. Pohl, F. Radlinski, and T. Joachims, "Recommending related papers based on digital library access records," in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, ACM, 2007, pp. 417-418.
[18] C. Yang *et al.*, "CARES: A ranking-oriented CADAL recommender system," in *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, ACM, 2009, pp. 203-212.
[19] K. Sugiyama and MY. Kan, "Scholarly paper recommendation via user's recent research interests," in *Proceedings of the 10th annual joint conference on Digital libraries*, ACM, 2010, pp. 29-38.
[20] M. Gori and A. Pucci, "Research paper recommender systems: A random-walk based approach," in *IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE, 2006, pp. 778-781.
[21] J. Jardine and S. Teufel, "Topical PageRank: A model of scientific expertise for bibliographic search," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 501-510.
[22] Y. Sun *et al.*, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," in *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992-1003, 2011.
[23] DA. Cohn and T. Hofmann, "The missing link-a probabilistic model of document content and hypertext connectivity," *Advances in neural information processing systems*, pp. 430-436, 2001.
[24] Q. Mei *et al.*, "Topic modeling with network regularization," in *Proceedings of the 17th international conference on World Wide Web*, ACM, 2008, pp. 101-110.
[25] J. Chang and D. Blei, "Relational topic models for document networks," *Artificial Intelligence and Statistics*, pp. 81-88, 2009.
[26] HD. White. "Authors as citers over time," *Journal of the Association for Information Science and Technology*, vol. 52, no. 2, pp. 87-108, 2001.
[27] N. Harwood. "Publication outlets and their effect on academic writers' citations," *Scientometrics*, vol. 77, no. 2, pp. 253, 2008.
[28] RN. Lichtenwalter, JT. Lussier, and NV. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 243-252.
[29] C. Wang and DM. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2011, pp. 448-456.
[30] AP. Dempster, NM. Laird, and DB. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1-38, 1977.
[31] N. Bassiou and C. Kotropoulos, "Rplsa: A novel updating scheme for probabilistic latent semantic analysis," *Computer Speech & Language*, vol. 25, no. 4, pp. 741-760, 2011.
[32] DR. Radev, P. Muthukrishnan, and V. Qazvinian, "The ACL anthology network corpus," *Language Resources and Evaluation*, vol. 47, no. 4, pp. 919-944, 2013.
[33] J. Tang *et al.*, "Arnetminer: Extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 990-998.
[34] MA. Turk and AP. Pentland, "Face recognition using eigenfaces," in *Proceedings of Computer Vision and Pattern Recognition*, 1991, pp. 586-591.
[35] SE. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, Springer-Verlag New York, Inc., 1994, pp. 232-241.
[36] Z. Nie *et al.*, "Object-level ranking: Bringing order to web objects," in *Proceedings of the 14th international conference on World Wide Web*, ACM, 2005, pp. 567-574.
[37] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2002, pp. 133-142.
[38] L. Guo *et al.*, "Exploiting fine-grained co-authorship for personalized citation recommendation." *IEEE Access*, vol. 5, pp. 12714-12725, 2017.
[39] D. Mu *et al.*, "Query-focused personalized citation recommendation with mutually reinforced ranking," *IEEE Access*, vol. 6, pp. 3107-3119, 2018.
[40] T. Dai *et al.*, "Explore semantic topics and author communities for citation recommendation in bipartite bibliographic network," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-19, 2017.
[41] X. Cai *et al.*, "Heterogeneous information network embedding based personalized query-focused astronomy reference paper recommendation," *International Journal of Computational Intelligence Systems*, vol. 11, pp. 591–599, 2018.
[42] S. Li *et al.*, "Conference paper recommendation for academic conferences," *IEEE Access*, vol. 6, pp. 17153-17164, 2018.
[43] JD. Mcauliffe and DM. Blei, "Supervised topic models," *Advances in neural information processing systems*, pp. 121-128, 2008.
[44] X. Cai *et al.*, "A three-layered mutually reinforced model for personalized citation recommendation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-12, 2018.
[45] R. Sharma, D. Gopalani, and Y. Meena, "Concept-based approach for research paper recommendation," in *International Conference on Pattern Recognition and Machine Intelligence*, Springer, Cham, 2017, pp. 687-692.

[46] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188-1196.
[47] S. Pan *et al.*, "Adversarially regularized graph autoencoder," *CoRR* abs/1802.04407, 2018.
[48] S. Pan *et al.*, "Tri-party deep network representation," *Network*, vol. 11, no. 9, pp. 1895-1901, 2016.
[49] S. Pan *et al.*, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 744-758, 2017.
[50] S. Pan *et al.*, "Joint structure feature exploration and regularization for multi-task graph classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 715-728, 2016.
[51] F. Xiong *et al.*, "Social recommendation with evolutionary opinion dynamics,". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, pp. 1-13, 2018.