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

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Conference Paper Recommendation for Academic Conferences

SHUCHEN LI[1], PETER BRUSILOVSKY[2], (Senior Member, IEEE), SEN SU[1], and XIANG CHENG[1]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15260, USA

ABSTRACT With the rapid growth of scientific publications, research paper recommendation which suggests relevant research papers to users can bring great benefits to researchers. In this paper, we focus on the problem of recommending conference papers to the conference attendees. While most of the related existing methods depend on the content-based filtering, we propose a unified recommendation method which exploits both the contents and the authorship information of the papers. In particular, besides the contents, we exploit the relationships between a user and a paper's authors for recommendation. In our method, we extract several features for a user-paper pair from the citation network, the coauthor network, and the contents, respectively. In addition, we derive a user's pairwise preference towards the conference papers from the user's bookmarked papers in each conference. Furthermore, we employ a pairwise learning to rank model which exploits the pairwise user preference to learn a function that predicts a user's preference towards a paper based on the extracted features. We conduct a recommendation performance evaluation using real-world data and the experimental results demonstrate the effectiveness of our proposed method.

INDEX TERMS Authorship Information, Citation Network, Coauthor Network, Learning to Rank, Paper Recommendation

I. INTRODUCTION

During the past few years, the number of academic publications has increased a lot and scholars are experiencing a troublesome information overload problem, i.e., there are an overwhelming amount of published papers in their research domain. Although some search engines can help researchers find relevant papers, they have to manually specify the search queries and identify the papers of interest from the search results. To tackle the information overload problem, research paper recommendation which aims to recommend papers of interest to users can bring great benefits to researchers. Nowadays, a variety of academic conferences have provided a great platform for scholars to publish and exchange their latest researches, which boosts the development and spread of new techniques. Also, some online social systems such as Conference Navigator 3 (CN3)[1] are employed to enhance conference attendees' experience at conferences. In CN3, users can browse the schedule of a conference's programs and events, read the metadata of each accepted paper, and bookmark the papers that they are interested in, which provides a great opportunity to study the paper recommendation problem. In this paper, we make a research on the problem of recommending accepted papers of a new conference to the conference attendees. Since these new conference papers have no user feedbacks and historical information, recommending such papers inherently faces a cold-start problem.

In the field of research paper recommendation, most of the related existing methods are mainly based on the content-based filtering [1], however, few of the studies have exploited the authorship information of the papers to make paper recommendation. Based on our observations, the relationships between a user and a paper's authors could also have a great impact on the user's interest towards the paper. For instance, if a paper comes from the authors who share the similar research interest with a user, the user may be more likely to be interested in the paper. Generally, users and authors can have three different types of relationships: citation relationships, coauthor relationships, and research interest correlations. As each user is also an author who has published several research papers, based on all the authors' publications and each publication's references, we can build

[1]http://halley.exp.sis.pitt.edu/cn3/portalindex.php

a citation network and a coauthor network among all the authors. Also, we can derive an author's research interest from the contents of the author's past publications. In this paper, we propose a paper recommendation method named $CPRec$ which exploits both the content information and the relationships between users and authors. To this end, given a user and a paper, we extract several features from three different types of information: the citation network, the coauthor network, and contents, respectively. In addition, to combine all the extracted features, we employ a pairwise learning to rank [2] model to learn a function which computes a user's preference towards a paper based on the extracted features. In detail, we propose to use a user's bookmarks in a specific conference to indict the user's preference towards all the papers in that conference. We assume that a user is more interested in the papers he/she has bookmarked than the rest papers in a conference, which forms a pairwise user preference. Moreover, we generate the training data utilizing all the user's pairwise preferences for model learning. After training the model, we learn a function which is used to predict a user's preference towards a paper based on the extracted features. Finally, given a new conference, we use the learned function to make personalized conference paper recommendation to the conference participants.

To sum up, the primary contributions of our research are listed as follows:

- We propose to exploit both the contents and the authorship information for conference paper recommendation. Specially, we extract three different types of features for a user-paper pair: citation-based features, coauthor-based features, and content-based features.
- We propose to utilize a user's bookmarks in a conference to derive the user's pairwise preference towards the papers in that conference.
- We employ a pairwise learning to rank model which adopts the pairwise user preference to learn a function that predicts a user's preference towards a paper based on the extracted features.
- We evaluate the performance of our method using real-world data collected from CN3. Experimental results show that our method is superior to some alternatives and the methods which consider only one single feature exploited in our method.

The rest of the paper is organized as follows. In Section II, we give a brief review of the related studies. Section III formally defines the conference paper recommendation problem. We describe the details of our recommendation model in Section IV. In Section V, we report the experimental results. Finally, we make the summarization and conclusions in Section VI.

II. RELATED WORK

In this section, we briefly review the related studies, including research paper recommendation and learning to rank techniques.

A. RESEARCH PAPER RECOMMENDATION

Due to its usefulness and importance, research paper recommendation has attracted many researchers' attention [1], [3]–[18]. Based on the different recommendation purposes, there are two main kinds of paper recommendation: one is to recommend interesting or relevant papers to the researchers [3]–[10], [15], [18] and the other one aims to recommend citations for a paper [11]–[14], [16], [17]. Our research is related to the former one, i.e., to recommend interesting papers of a conference to the conference participants. Among the related studies, Beel et al. [1] gave an overview on the published articles about research paper recommender systems. They found that most of the articles have applied content-based filtering, where TF-IDF [19] is the most adopted weighting scheme. Besides, they observed that the majority of the approaches need users to provide the input rather than inferring the user interest automatically. Sugiyama et al. [5] proposed a method which recommends research papers to a user by modeling the user's research interest through the user's past published papers. They enhanced a user's profile by combining the user's past publications with each publication's referenced papers as well as the papers that cite the publication. Later, they [3] proposed to exploit the citation network to find potential citation papers via collaborative filtering. Also, they investigated the different importance of the logical sections of a paper in representing the paper. In addition, Nascimentol et al. [10] presented a source independent framework for recommending scholarly papers, where they required users to provide one paper as the input and exploited available metadata (the title and abstract) to compute the similarities between the candidate papers and the input paper. In [7], they presented the paper recommender system in a literature management system called Docear, which uses mind maps for information management and also for recommendation.

The studies mentioned above mainly adopt content-based filtering, while there are some studies focusing on using collaborative filtering for publication recommendation. For instance, Yang et al. [9] presented a book recommender system which applies a ranking-oriented collaborative filtering method that exploits the data from users' access logs for recommendation. They computed the user similarity via the Average Precision correlation coefficients and used a random walk based algorithm for generating personalized recommendation results. In addition, some studies [4], [8] proposed a hybrid recommendation approach which combines content-based filtering and collaborative filtering. For example, Wang et al. [4] proposed a collaborative topic regression model which combines the traditional collaborative filtering with probabilistic topic modeling for recommending scientific articles.

Different from previous studies, in this paper, we propose to exploit both the contents and the relationships between a user and a paper's authors for paper recommendation, which have not yet been much explored. Moreover, we utilize a user's bookmarks in each conference to derive the user's pairwise preference and employ a supervised learning to rank model to learn a prediction function.

B. LEARNING TO RANK

Learning to rank is a popular kind of machine learning techniques used for learning a ranking model in a ranking-oriented task such as information retrieval, recommender systems, and computational advertising [2], [20]. In general, learning to rank approaches can be classified into three different types: pointwise, pairwise, and listwise.

The pointwise approaches directly apply the existing regression-based, classification-based, or ordinal regression-based machine learning methods to solve the ranking problem. They aim at predicting the accurate relevance score for each training object, while the ranking orders are ignored in these approaches.

Unlike the pointwise methods, the pairwise approaches care about the relative order of two objects. In pairwise approaches, the ranking problem is transformed into a classification or a regression problem on object pairs, i.e., to determine which object in the pair is ranked higher than the other one. There are many pairwise learning to rank approaches having been proposed, such as Ranking SVM [21], RankNet [22], and LambdaRank [23]. In Ranking SVM, a SVM is used for correctly classifying the order of objects pairs. RankNet adopts cross entropy loss as the loss function, which is an upper bound of the pairwise 0-1 loss. Also, it employs a neural network for learning and uses gradient descent for optimization.

Different from the aforementioned methods, the listwise approaches take the whole ranking lists as training instances in leaning and also predict a ranking list out. In this way, the complete ranking structure is kept and the loss functions can be directly related to the target evaluation measures such as MAP and NDCG [19]. For example, SVM MAP [24] directly optimizes for the MAP metric and uses a general SVM learning algorithm to fine a globally optimal solution.

In this paper, we adopt a pairwise learning to rank approach to learn a function which predicts a user's preference toward a paper.

III. PROBLEM DEFINITION

In this paper, conference paper recommendation aims to recommend accepted papers of a new conference to the attendees before the starting of the conference. As shown in Figure 1, let $U = \{u_1, u_2, ..., u_N\}$ and $C = \{c_1, c_2, ..., c_O\}$ denote the users and the past finished conferences, respectively. For each user $u_i \in U$, he/she has bookmarked some papers of a conference which he/she is interested in. Also, each user has published some scientific articles, therefore, a user is also an author. Sometimes, researchers may collaborate on some researches and coauthor several papers. Therefore, users can establish coauthor relationships. In addition, since a paper usually needs to cite some related research papers, users can build citation relationships through their publications indirectly. For each conference $c_i \in C$, it has a set of the accepted papers, denoted as D_{c_i}, and a list of attendees, denoted as $U_{c_i} \subseteq U$. Additionally, for each paper $d_j \in D_{c_i}$, it has an author list, denoted as A_{d_j}, and some content information such as a title and an abstract.

Given a new conference c_{new}, the accepted papers $D_{c_{new}}$ of this conference, and the conference attendees $U_{c_{new}}$, the conference paper recommendation problem can be formally defined as ranking all the papers in $D_{c_{new}}$ for each attendee in order to recommend top-k papers of interest to the attendees.

IV. RECOMMENDATION APPROACH

In this section, we will first present all the features exploited in the model and then introduce the details of our recommendation model. At last, we will describe the final conference paper recommendation step.

A. FEATURE EXTRACTION

To make paper recommendation, given a user u_i and a paper d_j, we generate a feature vector for model learning, denoted as $\Phi(u_i, d_j)$. In the field of research paper recommendation, most of the existing methods extract features from the contents [1], since the textual information is a direct and evident resource to indicate a user's research interest. However, few of studies have exploited the authorship information of the papers. Based on our observations, the relationships between a user and the authors of a paper also play an important role in influencing the user's interest towards the paper. Generally, users and authors can establish three different types of relationships: citation relationships, coauthor relationships, and research interest correlations. The citation relationships and coauthor relationships can be derived from users' publications, which form a citation network and a coauthor network of all the users. In addition, a user's research interest can be derived from the contents of the user's past publications. Therefore, in this paper, we extract features from three different types of resources: the citation network, the coauthor network, and contents, respectively. In the following, we will introduce each feature in detail.

1) Citation-based Features

A citation is an indicator to show a user's interest towards a paper or an author, therefore, citation relationships may be useful for paper recommendation. For instance, if a user has cited an author's papers many times before, he/she may be interested in the author's newly published papers in a conference. Figure 2 shows a toy example of a citation network. As shown in Figure 2, we build a matrix $T_{M\times M}$ to represent the citation relationships among all the authors, where M is the number of authors and T_{ij} counts the total times that author u_i has cited author u_j. Note that, since the citation relationship is directed, $T_{M\times M}$ is an asymmetric matrix. Let T_i^+ and T_i^- denote the i-th row and column of matrix T, respectively. T_i^+ represents how author u_i cites other authors and T_i^- denotes how author u_i is cited by other authors. Note that, a user is also an author in our problem. In the following, we extract several citation-based features for

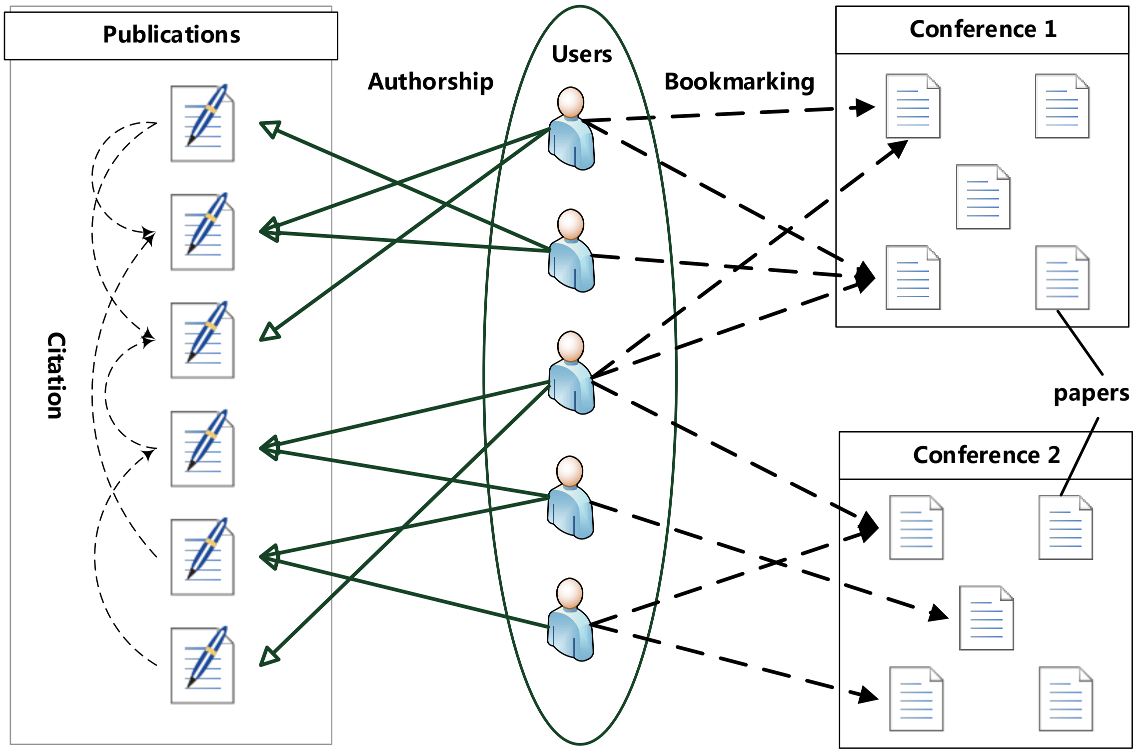


FIGURE 1: A system overview.

a user-paper pair (u_i, d_j) based on the citation relationships between user u_i and the authors of paper d_j.

Citation count. This feature records the maximal times that user u_i has cited each author of paper d_j, which represents the direct citation relationships between user u_i and the authors of paper d_j. We formally define this feature as below:

$$Ctt\_Count(u_i, d_j) = \max\{T_{ik} \ : u_k \in A_{d_j}\},$$

where A_{d_j} denotes all the authors of paper d_j.

Common neighborhoods. This type of feature counts the largest number of common neighbors who are shared between user u_i and the authors of paper d_j in the citation network. Since the citation relationship is directed, there are four different sets of neighbors between two users, which leads to four different features. Let out and in denote the outbound (i.e., citation) and inbound direction of the links in the citation network, respectively. These four features are defined as follows:

$$\begin{aligned} Ctt\_Com\_out\_out(u_i, d_j) &= \max\{|T_i^+ \cap T_k^+| \ : u_k \in A_{d_j}\}, \\ Ctt\_Com\_out\_in(u_i, d_j) &= \max\{|T_i^+ \cap T_k^-| \ : u_k \in A_{d_j}\}, \\ Ctt\_Com\_in\_out(u_i, d_j) &= \max\{|T_i^- \cap T_k^+| \ : u_k \in A_{d_j}\}, \\ Ctt\_Com\_in\_in(u_i, d_j) &= \max\{|T_i^- \cap T_k^-| \ : u_k \in A_{d_j}\}, \end{aligned}$$

where $| \cdot |$ denotes the size of the set.

Cosine similarities. This feature set computes the maximal cosine similarities between the citation links of user u_i and those of paper d_j's authors in the citation network, which takes the citation times into consideration. Similar to "Common neighborhoods", there are also four different combinations of the citation links and these features are defined as below:

$$\begin{aligned} Ctt\_Cos\_out\_out(u_i, d_j) &= \max\{cos(T_i^+, T_k^+) \ : u_k \in A_{d_j}\}, \\ Ctt\_Cos\_out\_in(u_i, d_j) &= \max\{cos(T_i^+, T_k^-) \ : u_k \in A_{d_j}\}, \\ Ctt\_Cos\_in\_out(u_i, d_j) &= \max\{cos(T_i^-, T_k^+) \ : u_k \in A_{d_j}\}, \\ Ctt\_Cos\_in\_in(u_i, d_j) &= \max\{cos(T_i^-, T_k^-) \ : u_k \in A_{d_j}\}, \end{aligned}$$

where $cos()$ denotes the cosine similarity.

2) Coauthor-based Features

Coauthor relationships represent the collaborations among different authors, which could also have an impact on a user's decision to bookmark a paper, e.g., a user may like the papers from his/her close collaborators. In Figure 3, we show a toy example of a coauthor network. Similar to the citation network, we also use a matrix $S_{M\times M}$ to represent the coauthor relationships among all the authors, where S_{ij} counts the times that author u_i and u_j have collaborated. However, different from the citation relationships, the coauthor relationships are undirected, therefore, S is a symmetric matrix. Let S_i denote the i-th row of matrix S and we define some coauthor-based features for a user-paper pair (u_i, d_j) based on the coauthor relationships between user u_i and the authors of paper d_j as follows.

Coauthor count. This feature captures the maximal times that user u_i have collaborated with each author of paper d_j, which is defined as below:

$$Coa\_Count(u_i, d_j) = \max\{S_{ik} \ : u_k \in A_{d_j}\}.$$

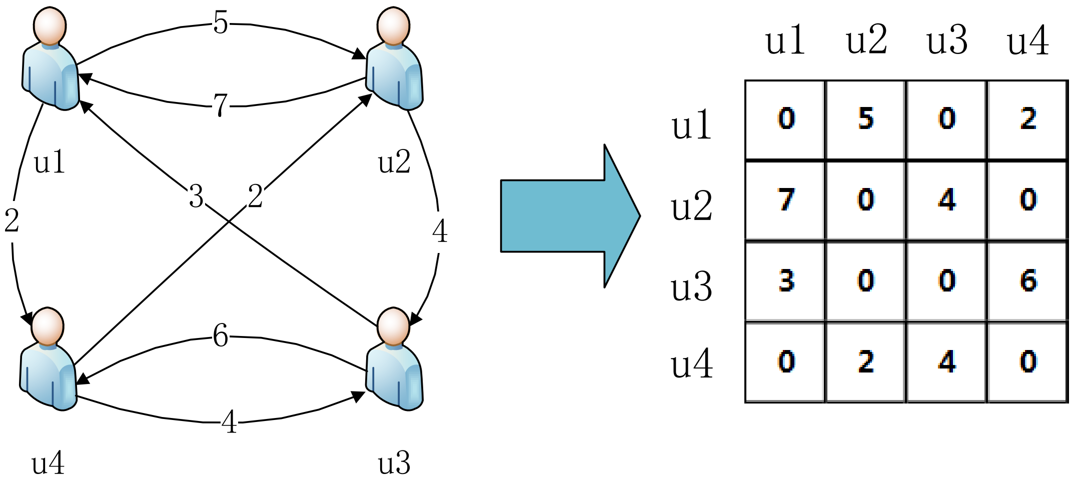


FIGURE 2: A toy example of a citation network. In the figure, a link represents a directed citation relationship between two users and the number attached in a link denotes the number of citations.

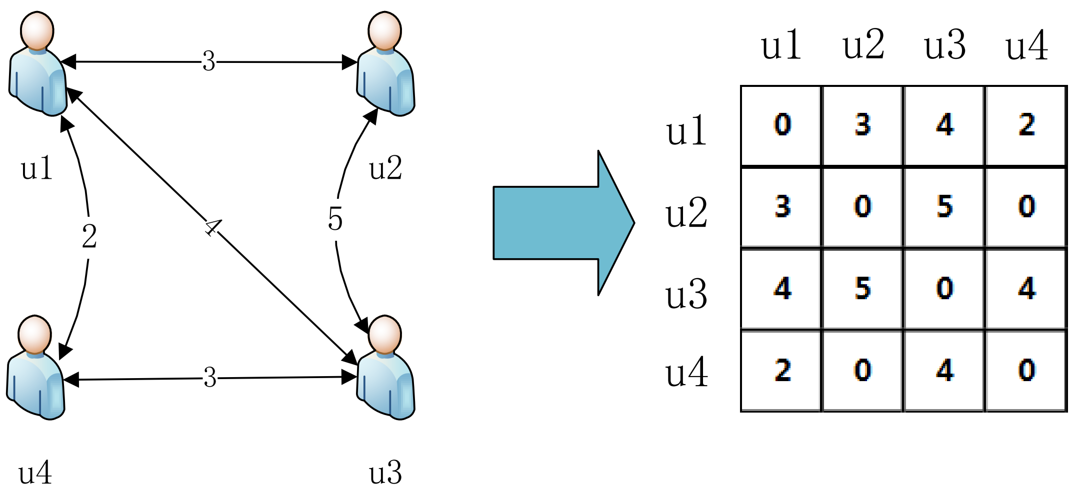


FIGURE 3: A toy example of a coauthor network. In the figure, a link represents an undirected coauthor relationship between two users and the number attached in a link denotes the number of collaborations.

Common coauthors. This feature counts the largest number of common coauthors shared between user u_i and the authors of paper d_j in the coauthor network. We define this feature as follows:

$$Coa\_Com(u_i, d_j) = \max\{|S_i \cap S_k| \ : u_k \in A_{d_j}\}.$$

Cosine similarity. This feature computes the maximal cosine similarity between the coauthor relationships of user u_i and those of paper d_j's authors, which takes the collaboration times into consideration and is defined as follows:

$$Coa\_Cos(u_i, d_j) = \max\{cos(S_i, S_k) \ : u_k \in A_{d_j}\}.$$

3) Content-based Features

Intuitively, the contents of a paper have a significant impact on attracting a user's attention, which is also the most widely exploited resource in research paper recommendation [1]. If a paper matches a user's research interest, it could be highly possible for the user to pay attention to this paper. However, besides using the contents of a paper, we can also exploit the research interests of the authors for recommendation. We assume that if a user and an author share the similar research interests, the user may be likely to be interested in the author's papers. To model a user's research interests, we propose to use the contents of the user's past published papers. In detail, we extract the words from some textual information of each paper, such as the title and the abstract. For each user u_i, we generate a user profile V_{u_i} by gathering all the words of user u_i's published papers together. A user profile V_{u_i} is described using a vector space model. Specifically, each V_{u_i} is represented by a fixed-size vector $(w_{1,i}, w_{2,i}, ..., w_{n,i})$, where each index denotes a word in the vocabulary and $w_{t,i}$ denotes the weight of word w_t in V_{u_i}. There are many ways to compute the weights of the words, in this paper, we adopt the most popular weighting scheme: TF-IDF [19]. Using TF-IDF, each weight is computed by the product of term frequency and inverse document frequency of each word. Analogously, for a paper d_j, we can also use a vector V_{d_j} to represent it. In the following, we extract two content-based features for a user-paper pair (u_i, d_j).

User-paper similarity. This feature computes the cosine similarity between user u_i's profile V_{u_i} and paper d_j's profile V_{d_j}, which measures how much paper d_j matches user u_i's research interests based on the paper contents. This is also the most typical content-based method for paper recommendation [1]. We formally define this feature as below:

$$Txt\_Paper(u_i, d_j) = cos(V_{u_i}, V_{d_j}).$$

User-author similarity. Unlike the feature "user-paper similarity", this feature computes the largest cosine similarity between user u_i's profile V_{u_i} and the profiles of paper d_j's authors, which can be regarded as a measure of the similarity between the user and the authors' research interests. This feature is defined as follows:

$$Txt\_Author(u_i, d_j) = \max\{cos(V_{u_i}, V_{u_k}) \ : u_k \in A_{d_j}\}.$$

B. RECOMMENDATION MODEL

As mentioned in previous sections, an attendee of a conference can bookmark the conference's papers that he/she is interested in. We consider that these bookmarks have a strong indication about a user's preference towards the papers in that conference, i.e., it indicates that a user is much more interested in the papers which he/she has bookmarked than the rest in the conference, which forms a pairwise user preference. To exploit this kind of pairwise user preferences, we adopt the pairwise learning-to-rank techniques to build the recommendation model. The basic idea of pairwise learning to rank approaches is trying to learn a classifier or a ranking function which can correctly classify the ranking orders of object pairs [2], [20].

In this paper, we employ a widely used pairwise learning to rank method called Ranking SVM [21], [25], which learns a function for pairwise classification. In the previous step, we generate a feature vector $\Phi(u_i, d_j)$ for a user u_i and a paper d_j. Then, we define a prediction function which predicts user u_i's preference towards paper d_j based on the extracted features as below:

$$f(u_i, d_j) = \langle \vec{w}, \Phi(u_i, d_j) \rangle \,,$$

where $\vec{w}$ is a weight vector and $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. Given a conference c_i and its accepted papers D_{c_i}, for each attendee u_i in the attendee list U_{c_i}, he/she may have a set of bookmarked papers, denoted as $P_{c_i}^{u_j}$, and the rest papers are denoted as $N_{c_i}^{u_j} = D_{c_i} - P_{c_i}^{u_j}$. We assume that user u_i prefers the papers in the $P_{c_i}^{u_j}$ to the papers in the $N_{c_i}^{u_j}$, which can be formally represented as follow:

$$f(u_i, d_j) - f(u_i, d_k) = \begin{cases} > 0 & if\ d_j \in P_{c_i}^{u_j}\ and\ d_k \in N_{c_i}^{u_j}, \\ < 0 & if\ d_j \in N_{c_i}^{u_j}\ and\ d_k \in P_{c_i}^{u_j}, \\ unknown & else. \end{cases} \tag{1}$$

	UMAP		HyperText	
	2009-2015	2016	2008-2015	2016
#Attendees	324	22	172	10
#Papers	990	129	688	67
#Bookmarks	3793	259	1541	60

TABLE 1: Some statistics of the pre-processed data.

Note that, in Equation (1), if paper d_j and paper d_k belong to the same set ($P_{c_i}^{u_j}$ or $N_{c_i}^{u_j}$), user u_i's preference towards paper d_j over paper d_k is undefined. Then, we generate the training samples as $\{((\Phi(u_i, d_j) - \Phi(u_i, d_k)), y_{ijk})\}$, where y_{ijk} is a label which equals to 1 if $d_j \in P_{c_i}^{u_j}$ and $d_k \in N_{c_i}^{u_j}$, and equals to -1 if $d_j \in N_{c_i}^{u_j}$ and $d_k \in P_{c_i}^{u_j}$. In this way, we transform the pairwise ranking problem into a classification problem. By gathering all training samples from every past conference, we construct the complete training data. Then, we learn a traditional SVM classifier to solve the classification problem. Formally, the formulation of Ranking SVM is defined as follows:

$$\begin{aligned} & minimize: \quad \tfrac{1}{2}\|\vec{w}\|^2 + C \cdot \textstyle\sum \xi_{ijk} \\ & subject\ to: \\ & \qquad y_{ijk} \cdot \langle \vec{w}, \Phi(u_i, d_j) - \Phi(u_i, d_k) \rangle \geq 1 - \xi_{ijk} \\ & \qquad \xi_{ijk} \geq 0 \end{aligned} \tag{2}$$

where ξ_{ijk} denotes the slack variable, $\| \cdot \|$ is the Frobenius norm, and C is a coefficient used to trade off between margin size and training error. Mathematically, Equation (2) is equivalent to the minimization of a regularized hinge loss function, which can be formally defined as below:

$$minimize \sum h(1 - y_{ijk} \cdot \langle \vec{w}, \Phi(u_i, d_j) - \Phi(u_i, d_k) \rangle) + \frac{1}{2C}\|\vec{w}\|^2$$

where $h(x)$ is the function $max(x, 0)$. To solve the optimization problem in Equation (2), we adopt the same solution described in the paper [25].

C. CONFERENCE PAPER RECOMMENDATION

In the final recommendation step, given the attendees $U_{c_{new}}$ and the accepted papers $D_{c_{new}}$ of a new conference c_{new}, for each attendee $u_i \in U_{c_{new}}$ and each candidate paper $d_j \in D_{c_{new}}$, we first extract the features for the user-paper pair (u_i, d_j). Then, using the prediction function $f(u_i, d_j)$ learned in the previous step, we can predict user u_i's preference towards the paper d_j based on the extracted features. After that, by ranking all the obtained attendee u_i's preferences towards the candidate papers, we can recommend top-k preferred conference papers to attendee u_i.

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) Dataset

To test the recommendation performance, we collect real-world data from Conference Navigator 3 (CN3). Specifically, we select two different conference series as the experimental subjects: UMAP (User Modeling, Adaptation and Personalization) and HyperText. For the UMAP conference series, we use the data of UMAP2016 as the test data and train our model based on the data collected from UMAP2009 to UMAP2015. Similarly, for the HyperText conference series, we use the data gathered from UMAP2008 to UMAP2015 for training and test on the data from HyperText2016. In detail, we collect the following information: (1) user information, including the user id and the user name; (2) conference information, including the conference id and the accepted papers; (3) paper information, including the paper id, the title, the authors, and the abstract; (4) each user's bookmarked papers in these conferences.

To collect users' publication information and the citation relationships among the publications, we use a public dataset named "DBLP-Citation-network V8"[2], published by Aminer [26]. This dataset contains 3,272,991 papers and 8,466,859 citation relationships. For each publication in this dataset, we collect the information such as the index, the title, the authors, the abstract, and the list of the references. In our experiments, we conduct a user matching between the CN3 system and the DBLP dataset, and find 8732 correctly matched users and authors in total.

Finally, for evaluation, we remove the users who don't have any bookmarks in these conferences or don't have any publications in the DBLP dataset. After preprocessing, some statistics of the data are shown in Table 1. On average, each attendee has about 12 bookmarks in the UMAP conference series and 9 bookmarks in the HyperText conference series.

2) Evaluation Metrics

We evaluate the performance of the methods in terms of top-k recommendation results. In the experiments, each method under comparison first computes a preference score for each candidate paper for a user and then recommends the top-k highest ranked papers to the user. To evaluate the accuracy of recommendation, we use four widely adopted metrics: $Precision@k$, $Recall@k$, F_1 score, and $MAP@k$ [27]. The $Precision@k$ measures the ratio of papers in the top-k recommendation list that are bookmarked by the user in the test data and the $Recall@N$ measures the ratio of bookmarked papers in the test data that are covered in the top-k recommendation list. The F_1 score is the harmonic average of the $Precision$ and $Recall$.

Given a user u_i in the test set, let $I_{u_i}^{test}$ denote the set of papers that are bookmarked by user u_i in the test data and $R_{u_i,k}$ denote the top-k papers that are recommended to user u_i. Then the $Precision@k$, $Recall@k$, and F_1 score are respectively defined as follows:

$$Precision@k = \frac{1}{|U|}\sum\nolimits_{u_i \in U} \frac{\left|R_{u_i,k} \cap I_{u_i}^{test}\right|}{k},$$
$$Recall@k = \frac{1}{|U|}\sum\nolimits_{u_i \in U} \frac{\left|R_{u_i,k} \cap I_{u_i}^{test}\right|}{\left|I_{u_i}^{test}\right|},$$
$$F_1@k = \frac{2 \times Precision@k \times Recall@k}{Precision@k + Recall@k},$$

where $|\cdot|$ is the size of the set. In addition, $MAP@k$ is the mean of all the users' Average Precision (AP) scores at position k, which takes the order of the recommendation results into consideration. The $AP@k$ score of each user and $MAP@k$ are defined as follows:

$$AP_{u_i}@k = \frac{\sum_{k=1}^{n} Precision@k \times I(k)}{|I_{u_i}^{test}|},$$
$$MAP@k = \frac{1}{|U|}\sum\nolimits_{u_i \in U} AP_{u_i}@k,$$

where $I(k)$ is an indicator function which equals to 1 if the paper at position k is in user u_i's bookmark list in the test set and equals to 0 otherwise. For convenience, in the following, we replace $Precision@k$ with $Pr@k$, $Recall@k$ with $Rc@k$.

3) Compared Methods

Since each feature exploited in our method $CPRec$ can be used to make paper recommendation independently as an unsupervised method, we devise the methods that exploit only one single feature. In total, there are 14 different methods and we name each of them as the name of the feature it uses. Note that, the method *Txt_Paper* is the traditional content-based paper recommendation using TF-IDF weighting scheme. Also, we devise a method named *LDA*, which employs a probabilistic topic model Latent Dirichlet Allocation (LDA) [28] to derive the topic distributions of the users and papers. Moreover, it uses the Jensen-Shannon divergence [29] to compute the distance between the topic distributions of a user and a paper for recommendation. In addition, we compare with a state-of-the-art content-based paper recommendation method [3], which is an extension of the research [5]. This method uses collaborative filtering to find potential citations publications for each paper and enhances each paper's profile by incorporate the information from the papers that cite it, its direct citation papers, and its potential citation papers. In this paper, we named this method as "*SRC*".

4) Parameter Setting

For our method *CPRec*, we empirically set the hyperparameter C in Equation (2) to 0.01. For the method *LDA*, by tuning the topic number, we find it reaches the best performance when topic number is 30. The other hyperparameters of *LDA* are set as the default values in Gensim[3]. Besides, the parameters of the method *SRC* are set according to the paper [3].

B. EXPERIMENTAL RESULTS

1) Top-5 Recommendation Performance Comparison

In this section, we analyze the top-5 recommendation performance of all the methods under the metrics of *Precision*, *Recall*, and F_1 score. The results are presented in Table 2, we can clearly observe that our proposed method $CPRec$ always

[2]https://www.aminer.cn/citation

[3]https://radimrehurek.com/gensim/

Type	Methods	UMAP2016			HyperText2016		
		$Pr@5$	$Rc@5$	$F_1@5$	$Pr@5$	$Rc@5$	$F_1@5$
Citation	Ctt_Count	0.245455	0.146815	0.183733	0.18	0.119444	0.143599
	Ctt_Com_out_out	0.254545*	0.194532*	0.220529*	0.22	0.153333	0.180714
	Ctt_Com_out_in	0.227273	0.167733	0.193015	0.16	0.087222	0.112899
	Ctt_Com_in_out	0.236364	0.133439	0.170578	0.22	0.147778	0.176798
	Ctt_Com_in_in	0.254545*	0.149077	0.188032	0.22	0.163889	0.187844
	Ctt_Cos_out_out	0.227273	0.178875	0.200190	0.28*	0.201667*	0.234464*
	Ctt_Cos_out_in	0.245455	0.193975	0.216699	0.22	0.150556	0.178771
	Ctt_Cos_in_out	0.236364	0.127831	0.165925	0.26	0.181667	0.213887
	Ctt_Cos_in_in	0.227273	0.137227	0.171127	0.26	0.200556	0.226441
Coauthor	Coa_Count	0.236364	0.083153	0.123025	0.14	0.082778	0.104040
	Coa_Com	0.327273†	0.223688†	0.265743†	0.24	0.183889	0.208231
	Coa_Cos	0.318182	0.209348	0.252538	0.26†	0.203889†	0.228551†
Content	Txt_Paper	0.227273	0.105215	0.143840	0.26	0.213889	0.234701
	Txt_Author	0.272727#	0.201196#	0.231563#	0.28#	0.225000#	0.249505#
	LDA	0.163636	0.112547	0.133367	0.24	0.178333	0.204622
	SRC	0.245454	0.173205	0.203096	0.26	0.219778	0.238203
	CPRec	**0.354545**	**0.254223**	**0.296118**	**0.34**	**0.263333**	**0.296796**

TABLE 2: The top-5 recommendation performance comparison. The values with a superscript like *, †, or # denote the best performance among citation-based methods, coauthor-based methods, and content-based methods, respectively.

outperforms all the baseline methods cross all the metrics in both the conference series significantly. For instance, compared to the traditional TF-IDF based method *Txt_Paper*, *CPRec* improves the $F_1@5$ performance by about 105.9% in UMAP2016 and 26.5% in HyperText2016, respectively. Besides, *CPRec* performs much better than any method that uses only one single feature, which demonstrates the strength of combining the features together.

Among all the methods that exploit the features extracted from the citation network, method *Ctt_Com_out_out* and *Ctt_Cos_out_out* performs the best in UMAP2016 and HyperText2016, respectively, which indicates that a user is more likely to be interested in the paper from the authors who share many overlapped cited nodes with the user in the citation network. While, the performance of method *Ctt_Count* is always the worst in both conferences, which shows that a user may have less interests to bookmark the paper whose authors are the direct neighbors of the user in the citation network.

With regard to the methods using the features derived from the coauthor network, we find that method *Coa_Count* always performs the worst in both conferences, which could be due to the reason that if two users have a strong collaborative relationship, they may have already been familiar with each other's researches very well, therefore, they would be less motivated to bookmark their collaborators' papers. In addition, method *Coa_Com* performs the best in UMAP2016, while method *Coa_Cos* surpasses others in HyperText2016. Note that, in UMAP2016, some coauthor-based methods such as *Coa_Com* and *Coa_Cos* perform even better than the best content-based method, which manifests the potential power of utilizing the relationships between users and authors for paper recommendation.

Within all the methods that exploit content-based features, we observe that by considering the similarities between users and authors rather than users and papers, the recommendation performance gets promoted. As an example, in UMAP2016, the $F_1@5$ performance of method *Txt_Author* is about 61% higher than that of method *Txt_Paper*. Another finding is that using topic models to model the users and contents may hurt the recommendation performance in our problem. As shown in Table 2, method *LDA* generally performs worse than the TF-IDF based method *Txt_Paper*. Besides, method *SRC* always performs better than method *Txt_Paper*, which may due to that it incorporates much information from the relevant papers of a target paper for recommendation.

2) MAP@k Recommendation Performance Comparison

In this section, we evaluate and compare the MAP@k performance of all the methods in top-k recommendation. Unlike $Precision$, $Recall$, and F_1 score, the MAP metric takes the ranking order into consideration, which is suitable to measure the ranking performance.

Table 3 shows the MAP@5 and MAP@10 performance of all the methods in both conferences. We can observe that our method *CPRec* still achieves the best performance among all the comparison methods, which verifies the effectiveness of our proposed method in conference paper recommendation. With regard to the MAP@5 metric, *CPRec* outperforms the best baseline method in each conference (*Coa_Cos* in UMAP2016 and *Txt_Author* in HyperText2016) by about 10.6% and 19.1%, respectively.

Among all the content-based methods, method *Txt_Author* always outperforms the rest that exploit the content information of the candidate papers for recommendation, which demonstrates the usefulness of the authorship information in conference paper recommendation. In UMAP2016, method *Txt_Author* performs 76.6% and 90.4% better than method *Txt_Paper* under MAP@5 and MAP@10, respectively. Also, the performance of method *LDA* is much inferior to that of method *Txt_Paper*, which again shows the ineffectiveness of topic models in our problem.

		UMAP2016		HyperText2016	
Type	**Methods**	$MAP@5$	$MAP@10$	$MAP@5$	$MAP@10$
Citation	Ctt_Count	0.240202	0.192336	0.148333	0.116667
	Ctt_Com_out_out	0.291515*	0.250786*	0.178333	0.172376
	Ctt_Com_out_in	0.262576	0.243773	0.119667	0.110278
	Ctt_Com_in_out	0.236515	0.177674	0.200000	0.168948
	Ctt_Com_in_in	0.247424	0.191471	0.218333	0.187438
	Ctt_Cos_out_out	0.271970	0.219842	0.240333*	0.205022*
	Ctt_Cos_out_in	0.248232	0.206710	0.190000	0.194910
	Ctt_Cos_in_out	0.245000	0.180477	0.220333	0.187944
	Ctt_Cos_in_in	0.240758	0.191762	0.236333	0.203103
Coauthor	Coa_Count	0.196667	0.140221	0.119000	0.093056
	Coa_Com	0.340505	0.264792	0.210333	0.189333
	Coa_Cos	0.343636†	0.277437†	0.235333†	0.220075†
Content	Txt_Paper	0.182424	0.147050	0.249000	0.237999
	Txt_Author	0.322121#	0.280042#	0.260667#	0.251833#
	LDA	0.114394	0.110684	0.182667	0.164795
	SRC	0.244217	0.199523	0.254333	0.243943
	CPRec	**0.380152**	**0.317859**	**0.310333**	**0.273358**

TABLE 3: The $MAP@k$ recommendation performance comparison. The values with a superscript like *, †, or # denote the best performance among citation-based methods, coauthor-based methods, and content-based methods, respectively.

Within all the methods based on the citation network, method *Ctt_Com_out_out* performs the best in UMAP2016, while method *Ctt_Cos_out_out* outperforms other methods in HyperText2016, which shows the importance of the out-bound citation links in the citation network for paper recommendation. For coauthor-based methods, method *Coa_Cos* always performs better than others cross all the tests, which indicates that a user may pay more attention to the papers come the authors who have many overlapped coauthors with the user. Based on the performance of methods which exploit the features extracted from the citation network or the coauthor network, we observe that a user is more interested in the papers from the user's second next-hop nodes rather than the user's direct neighbor nodes in both the citation network and the coauthor network.

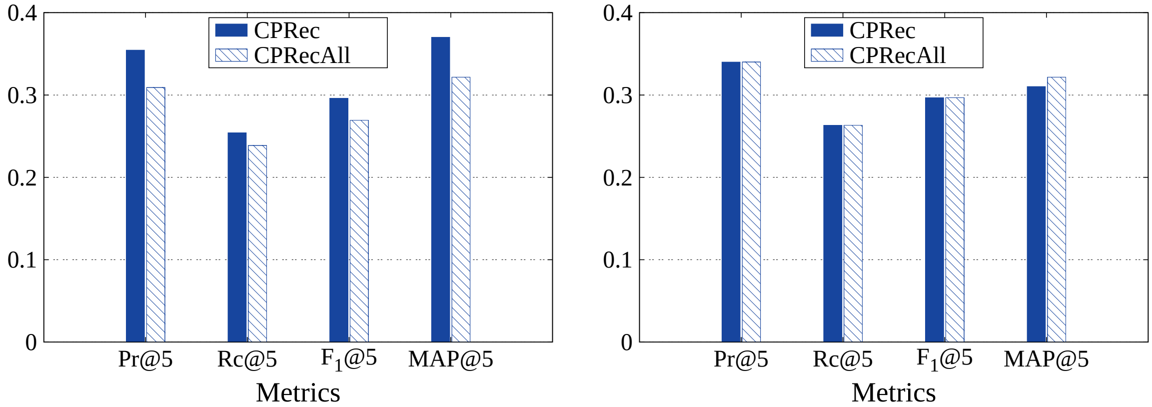
3) Different Training Strategies Comparison

In the previous experiments, we make paper recommendation in each conference series separately, i.e., we train a separate model for a different conference series. However, it remains a question what if we use the data from all the conference series to train a unified model. To verify it, we collect all the data from UMAP2009-2015 and HyperText2008-2015 to train a single model and then use this model to make separated predictions in UMAP2016 and HyperText2016. In this paper, we name this new method as *CPRecAll*. Figure 4 shows the top-5 recommendation performance comparison results of *CPRec* and *CPRecAll*. We can observe that method *CPRecAll* performs averagely 10% worse than method *CPRec* in UMAP2016, while, in HyperText2016, these two methods achieve much similar performance. Therefore, the results indicate that combining the data from different conference series doesn't improve the paper recommendation in each individual conference series, which may be due to the reason that each conference series has its own characteristic and different types of attendees.

VI. SUMMARY AND CONCLUSIONS

In this paper, we study the problem of recommending accepted papers of a new conference to the conference attendees. We propose a unified recommendation model which takes both the textual information and the relationships between a user and a paper's authors into consideration. In particular, we exploit three different types of relationships: citation relationships, coauthor relationships, and research interest correlations. The coauthor relationships and citation relationships are extracted from all the authors' publications and each publication's references. Besides, we derive an author's research interest from the contents of the author's past publications. In our method, we extract several features for a user-paper pair based on the citation network, the coauthor network, and the contents, respectively. In addition, we propose to derive a user's pairwise preference towards all the papers in a conference from the user's bookmarks in the conference. Furthermore, we employ a pairwise learning to rank model which exploits the pairwise user preference to learn a prediction function that computes a user's preference towards a paper based on the extracted features. Finally, we utilize the learned function to make personalized conference paper recommendation. We conduct extensive experiments using real-world data collected from Conference Navigator 3. The experimental results show that our proposed model outperforms all the compared methods significantly.

In this research, we emphasize the relationships between a user and a paper's authors in conference paper recommendation, which few of the related studies have exploited. The experimental results demonstrate that these relationships can be exploited to make effective personalized paper recommendation. It also verifies that, in addition to a paper's contents, the authorship information of a paper could have a great impact on a user's interest towards a paper as well. By utilizing both the contents and the authorship information, we achieve a better paper recommendation performance. Moreover, we



(a) UMAP2016

(b) HyperText2016

FIGURE 4: Top-5 performance comparison between *CPRec* and *CPRecAll*.

find that a user's bookmarks are a good resource to indicate the user's pairwise preference in each conference, i.e., a user prefers the papers which he/she has bookmarked to the rest papers in a conference. The experimental results show that these pairwise user preferences are very useful for conference paper recommendation.

However, in our method, we extract several features from the citation network and the coauthor network only based on the neighborhood structure. In our future work, we would like to employ some other techniques such as graph embedding to model the citation relationships and the coauthor relationships among the authors. Besides, we plan to investigate the time factor in paper recommendation, such as the temporal changes in a user's research interests.

REFERENCES

[1] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: a literature survey," Int. J. on Digital Libraries, vol. 17, no. 4, pp. 305–338, 2016.

[2] T. Liu, Learning to Rank for Information Retrieval. Springer, 2011.

[3] K. Sugiyama and M. Kan, "Exploiting potential citation papers in scholarly paper recommendation," in 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13, Indianapolis, IN, USA, July 22 - 26, 2013, 2013, pp. 153–162.

[4] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011, 2011, pp. 448–456.

[5] K. Sugiyama and M. Kan, "Scholarly paper recommendation via user's recent research interests," in Proceedings of the 2010 Joint International Conference on Digital Libraries, JCDL 2010, Gold Coast, Queensland, Australia, June 21-25, 2010, 2010, pp. 29–38.

[6] H. Xue, J. Guo, Y. Lan, and L. Cao, "Personalized paper recommendation in online social scholar system," in 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, Beijing, China, August 17-20, 2014, 2014, pp. 612–619.

[7] J. Beel, S. Langer, M. Genzmehr, and A. Nürnberger, "Introducing docear's research paper recommender system," in 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13, Indianapolis, IN, USA, July 22 - 26, 2013, 2013, pp. 459–460.

[8] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl, "Enhancing digital libraries with techlens+," in ACM/IEEE Joint Conference on Digital Libraries, JCDL 2004, Tucson, AZ, USA, June 7-11, 2004, Proceedings, 2004, pp. 228–236.

[9] C. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang, "CARES: a ranking-oriented CADAL recommender system," in Proceedings of the 2009 Joint International Conference on Digital Libraries, JCDL 2009, Austin, TX, USA, June 15-19, 2009, 2009, pp. 203–212.

[10] C. Nascimento, A. H. F. Laender, A. S. da Silva, and M. A. Gonçalves, "A source independent framework for research paper recommendation," in Proceedings of the 2011 Joint International Conference on Digital Libraries, JCDL 2011, Ottawa, ON, Canada, June 13-17, 2011, 2011, pp. 297–306.

[11] T. Ebesu and Y. Fang, "Neural citation network for context-aware citation recommendation," in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017, 2017, pp. 1093–1096.

[12] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, "On the recommending of citations for research papers," in CSCW 2002, Proceeding on the ACM 2002 Conference on Computer Supported Cooperative Work, New Orleans, Louisiana, USA, November 16-20, 2002, 2002, pp. 116–125.

[13] W. Huang, Z. Wu, L. Chen, P. Mitra, and C. L. Giles, "A neural probabilistic model for context based citation recommendation," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., 2015, pp. 2404–2410.

[14] Q. He, J. Pei, D. Kifer, P. Mitra, and C. L. Giles, "Context-aware citation recommendation," in Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010, 2010, pp. 421–430.

[15] N. Y. Asabere, F. Xia, Q. Meng, F. Li, and H. Liu, "Scholarly paper recommendation based on social awareness and folksonomy," IJPEDS, vol. 30, no. 3, pp. 211–232, 2015. [Online]. Available: https://doi.org/10.1080/17445760.2014.904859

[16] 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. 12 714–12 725, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2721934

[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, 2015. [Online]. Available: https://doi.org/10.1109/ACCESS.2015.2481320

[18] F. Xia, H. Liu, I. Lee, and L. Cao, "Scientific article recommendation: Exploiting common author relations and historical preferences," IEEE Trans. Big Data, vol. 2, no. 2, pp. 101–112, 2016. [Online]. Available: https://doi.org/10.1109/TBDATA.2016.2555318

[19] G. Salton and M. McGill, Introduction to Modern Information Retrieval. McGraw-Hill Book Company, 1984.

[20] H. Li, "A short introduction to learning to rank," IEICE Transactions, vol. 94-D, no. 10, pp. 1854–1862, 2011.

[21] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in Advances in Large Margin

Classifiers, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, pp. 115–132.

[22] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender, "Learning to rank using gradient descent," in Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005, 2005, pp. 89–96.

[23] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006, 2006, pp. 193–200.

[24] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007, 2007, pp. 271–278.

[25] T. Joachims, "Optimizing search engines using clickthrough data," in Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada, 2002, pp. 133–142.

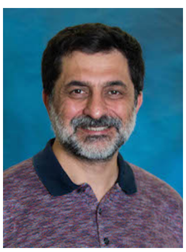
[26] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008, 2008, pp. 990–998.

[27] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Recommender Systems Handbook. Springer, 2011.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[29] C. D. Manning and H. Schütze, Foundations of statistical natural language processing. MIT Press, 2001.

SHUCHEN LI is pursuing the Ph.D. degree at Institute of Network Technology, Beijing University of Posts and Telecommunications, China. His research interests include recommender systems, machine learning, and artificial intelligence.



PETER BRUSILOVSKY received his BS (1983) and PhD (1987) degrees from Moscow "Lomonosov" State University. He received postdoctoral training at University of Sussex, University of Trier, and Carnegie Mellon University. He is currently a Professor of Information Science and Intelligent Systems at the University of Pittsburgh, where he directs Personalized Adaptive Web Systems (PAWS) lab. Peter Brusilovsky has been working in the field of adaptive educational systems, user modeling, and intelligent user interfaces for more than 20 years. He published numerous papers and edited several books on adaptive hypermedia, adaptive educational systems, user modeling, and the adaptive Web. Peter is the Editor-in-Chief of IEEE Transactions on Learning Technologies and a board member of several journals including User Modeling and User Adapted Interaction, ACM Transactions on the Web, and Web Intelligence and Agent Systems. He is a Senior Member of IEEE.

SEN SU received the Ph.D. degree in 1998 from University of Electronic Science and Technology, Chengdu, China. He is currently a Professor at the Beijing University of Posts and Telecommunications. His research interests include data privacy, cloud computing and internet services.

XIANG CHENG received the Ph.D. Degree in Computer Science from Beijing University of Posts and Telecommunications, China, in 2013. He is currently an Associate Professor at the Beijing University of Posts and Telecommunications. His research interests include information retrieval, data mining, and data privacy.