World Wide Web
https://doi.org/10.1007/s11280-019-00687-9

A novel hybrid publication recommendation system using compound information

Qiang Yang[1,2] · Zhixu Li[1,3] · An Liu[1] · Guanfeng Liu[1,4] · Lei Zhao[1] · Xiangliang Zhang[2] · Min Zhang[1] · Xiaofang Zhou[1,5]

Received: 6 December 2017 / Revised: 16 April 2019 / Accepted: 29 April 2019 /
Published online: 17 May 2019

Abstract
Publication recommendation is an interesting but challenging research problem. Most existing studies only use partial information of papers' contents, reference network or co-author relationship, which leads to an unsatisfied recommendation result. In this study, we propose a novel hybrid publication recommendation approach using compound information which retrieves top-K most relevant papers from a publication depository for a set of user input keywords. Our advantages comparing to the existing methods include: (1) Reaching a better recommendation results by taking the advantages of both content-based recommendation and citation-based recommendation and exploring much richer information of papers in one method; (2) Effectively solving the cold-start problem for new published papers by considering the vitality of papers and the impact factor of venues into the citation network; (3) Saving a large overhead in calculating the content-based similarity between papers and user input keywords by doing paper clustering based on the citation network. Extensive experiments on DBLP and Microsoft Academic datasets demonstrate that PubTeller improves the state-of-the-art methods with 4% in Precision and 4.5% in Recall.

Keywords Publication recommendation · Compound information · Edge-reinforced citation network · Citation network cluster

1 Introduction

Publication recommendation has been studied for decades [2, 32], which targets at recommending relevant papers to user's needs for reference. Although some academic search engines such as Google Scholar[1] can effectively help users find papers according to their

[1] http://scholar.google.com

This article belongs to the Topical Collection: *Special Issue on Web and Big Data*
Guest Editors: Junjie Yao, Bin Cui, Christian S. Jensen, and Zhe Zhao

✉ Zhixu Li
zhixuli@suda.edu.cn

Extended author information available on the last page of the article.

input keywords and constraints, the returned results can not always meet users' requirements due to the difficulties in understanding user needs as well as the fast increase of the publication quantity in recent years.

There have been a lot of efforts on publication recommendation [2]. The mainstream methods find the most relevant papers to the input keywords according to their relevance on contents (including title, keyword, abstract and sometimes the full paper). The relevance was firstly measured with traditional Information Retrieval techniques, and then improved with topic models [1, 34]. However, since there are always a large number of papers sharing the same hot topic, the top-K recommendation results based on paper contents only usually do not have a high precision. As a complement, some other works use the citation relations between papers for recommendation [32], which tend to give a higher ranking score to papers that are cited more by the others in the recommendation results. The citation score of a paper is not just decided by its frequency, but also the scores of papers that have cited the paper, thus some algorithms such as Random Walk have been used to calculate the ranking score of papers based on the citation network of publications [7, 8, 19, 32]. Some recent works also improve the citation-based methods by putting softly clustered papers into interest groups [31], or developing a multi-layer neural network probabilistic model to learn the semantic representations of citation contexts and cited papers [9]. However, the methods based on citations may easily recommend us some old papers that were cited a lot in history but already became less popular in recent years, and ignore some new papers that might be cited relatively less in total but were actually very popular in recent years. Another line of methods based on academic social network [18], i.e., co-author network, prefer to recommend papers sharing the same co-authors with the one(s) a user interests at. But this kind of methods may neglect some important papers that were written by some researches who seldom co-author with the others (i.e., some isolated nodes in the co-author network graph). In addition, some paper recommendation systems are developed [4, 7, 16, 38] which combine some of these approaches to recommend the suitable papers to users.

In this paper, we propose *PubTeller*, a novel hybrid publication recommendation approach using compound information. Different from previous approaches that only use one aspect of information, we use much richer information including not only paper contents and citation network of papers, but also the impact factor of venues (journals or conferences) and the vitality of papers according to the distribution of each paper's citation times from its published year to the current year. Intuitively, users usually prefer papers from top venues more than those from non-first-class venues; and prefer "up-to-date" papers rather than "out-of-date" papers.

The basic workflow of PubTeller is described in Figure 1: the input of the system includes a publication depository and a set of keywords that can describe the reading interests of a particular user, while the output is a set of top-K most relevant papers to the input keywords that are found from the publication depository. The core module is the paper ranking module which calculates the ranking score for each paper in the publication depository according to their relevance to the input keywords. Here we integrate the two mainstream recommendation methods, i.e., content-based recommendation and citation-based recommendation in a natural way. In the beginning, we propose a so-called edge-reinforced citation network which involves the vitality of papers and the impact factor of venues for solving the cold-start problem for recommending new papers, and then use a novel clustering algorithm for putting papers of similar topic into one cluster based on this edge-reinforced citation network. By analyzing these clusters with topic models, we find top-K most relevant clusters to the input keywords based on their similarities on topics and contents. We then find

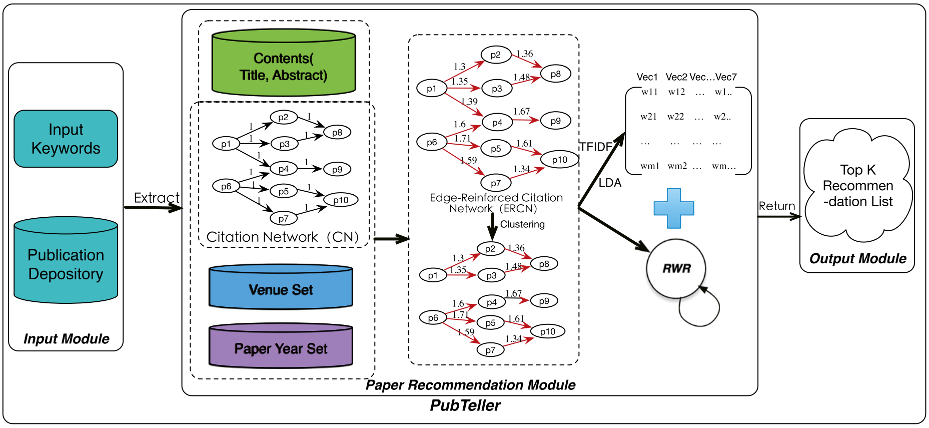


Figure 1 The workflow of the pubteller recommendation approach

top-K most popularly cited papers from each of the K clusters, and finally we can identify the top-K most relevant papers to the input keywords from the collected K lines of papers according to the topical and content similarity between each paper and the input keywords.

Compared with the existing approaches, PutTeller has the following three advantages: (1) Reaching a better recommendation results by taking the advantages of both content-based recommendation and citation-based recommendation and exploring much richer information of papers in one method; (2) Effectively solving the cold-start problem for new published papers by considering the vitality of papers and the impact factor of venues into the citation network; (3) Saving a large overhead in calculating the content-based similarity between papers and user input keywords by doing paper clustering based on the citation network.

Contributions Our contributions can be summarized as follows:

- We propose a compound paper recommendation approach exploring richer information of papers than previous methods for better recommendation.
- We propose a novel edge-reinforced citation network which involves the vitality of papers and the impact factor of venues for solving the cold-start problem for recommending new papers.
- We develop a novel clustering method for putting papers of similar topic into one cluster based on the edge-reinforced citation network, such that we can save a large overhead in calculating the content-based similarity between papers and user input keywords.

Extensive experiments conducted on DBLP and Microsoft Academic datasets demonstrate that PubTeller improves the state-of-the-art methods with 4% in Precision and 4.5% in Recall.

Roadmap The rest of the paper is organized as follows: We give the problem formulation in Section 2, and present our PubTeller algorithm in Section 3, and next report our experimental study in Section 4. After covering the related work in Section 5, we conclude the paper in Section 6.

2 Preliminary and problem formulation

In this section, we first list notations we will use in the rest of this paper and then formulate our problem. Table 1 summarizes some important notations to be used throughout this paper. Besides, a citation network will be built for developing our approach, where exists citation relation between two papers, i.e., “p_x cites p_y” for instance. In addition, when employing content of papers to measure the similarity between papers, we use Cosine Similarity function to compute the similarity between corresponding word vectors (say $\mathbf{p_i}$ and $\mathbf{p_j}$), where $dis(p_i, p_j) = Cos(\mathbf{p_i}, \mathbf{p_j}) = \frac{\mathbf{p_i} \cdot \mathbf{p_j}}{||p_i|| \cdot ||p_j||}$.

In this paper, the paper recommendation problem can be formally defined as follows:

Definition 1 Let $\mathcal{P} = \{p_1, p_2, ..., p_n\}$ denote the set of publications in the publication depository. For a set of keywords $\mathcal{KW} =< kw_1, kw_2, ..., kw_m >$ inputted by a user for one search, the task of the paper recommendation is to find from $\mathcal{P}$ the top-K most relevant papers to the keywords $\mathcal{KW}$ according to some predefined measure functions and constraints.

The PubTeller system takes a publication depository and a set of keywords that users are interested in as the inputs and it outputs a set of top-K most relevant papers to the input keywords that are found from the publication depository. The core module is the paper ranking module which calculates the ranking score for each paper in the publication depository according to their relevance to the input keywords. In details, it consists of the following four parts: contents of papers, the venue information, paper publishing year and the citation network where contents are used to build word vectors of paper and the last three components are utilized to construct the Edge-reinforced Citation Network. The proposed network takes consideration of the vitality of papers and the impact factor of venues and citation network simultaneously which can recommend more relevant articles to users. So as to improve the efficiency of our model, an effective clustering approach is proposed which

Table 1 Notations

Symbol	Description
$\mathcal{P}$	The paper set $\mathcal{P} = \{p_1, p_2, ..., p_M\}$
$\mathcal{KW}$	A set of input keywords shown as $\{kw_1, kw_2, ..., kw_m\}$
p	A paper
$v(p)$	A venue that a paper p published on
$y(p)$	The year that p published in
$c(p, y)$	the number of times that p is cited in its y-th year since its publication
Y	A year we compute the impact factor
G	A citation graph built in a corpus, where each node $p \in V$ is a paper, and each edge $\varepsilon \in E$ is a citation link.
c	A sub citation network in G
x_i	The center of a sub citation network c
$word_i$	A word of the publication depository
w_i	The weight of the word $word_i$
$t(p)$	A topic of paper p

can greatly reduce the unnecessary comparison between input information and candidate papers.

3 Recommendation algorithm

The basic idea and workflow of the algorithm have been briefly given in the Introduction. In this section, we first introduce the edge-reinforced citation network in Section 3.1, and then present the novel clustering algorithm for putting papers of similar topic into one cluster in Section 3.2. We finally present how we generate the top-K most relevant papers for given input keywords in Section 3.3.

3.1 Building edge-reinforced citation network (ERCN)

In order to let some newly published good papers on top venues could have a better chance to be recommended, we propose to build a edge-reinforced citation network by embedding the impact factor of venues (on which papers were published) as well as the vitality of papers into the citation network.

1) **Impact Factor and Paper Vitality Estimation:** We adopt the well-known SCI impact factor calculation method [2] to calculate the impact factor of venues. Note that the impact factor of a venue is changing year by year, therefore for a paper p published on a venue v, we should use the impact factor of v in the published year of p for a reference to potentially indicate the influence of the paper. Particularly, the impact factor of a journal in a specific year is the number of citations, received in that year, to the articles that are published in that journal during the two preceding years, divided by the total number of articles published in that journal during the two preceding years. More specifically, we calculate the impact factor of v for the given year Y as follows:

$$IF(v, Y) = \sum_{p \in \mathcal{P}, p \hookrightarrow v} \frac{C(p, Y-1, Y-2)}{N(p, Y-1, Y-2)} \tag{1}$$

where $p \hookrightarrow v$ denotes that the paper p is published on v and Y is the year we compute the impact factor of v. $C(p, Y-1, Y-2)$ gets the total citation times of p in year $Y-1$ and $Y-2$. $N(p, Y-1, Y-2)$ is the boolean function indicating if p is published by v in year $Y-1$ or $Y-2$, $N(p, Y-1, Y-2) = 1$; otherwise 0.

The vitality of a paper can be roughly reflected by the "age" of the paper as well as its citation times all through these years. Basically, the younger a paper is, the larger vitality degree it possesses. Besides, with the increasing of citation number of paper p, its vitality improves. More specifically, we estimate the vitality of a paper p with the following formulation:

$$V(p) = \frac{1}{1 + e^{-\sum_{y=1}^{Age(p)} - ln(1 - \frac{C(p,y)}{\sum_{p \neq p_j, x=1}^{Age(p_j,x)} C(p_j,x)})}} \cdot \{\frac{\sum_{y=1}^{Age(p)} C(p, y)}{\sum_{z=1, p \neq p_k}^{Age(p_k,z)} C(p_k, z)}\}^{\frac{1}{y(p)}} \tag{2}$$

where $C(p, y)$ is the citation number of paper p in its y-th year since its publication date and $y(p)$ gets the publication year of paper p, and $Age(\cdot)(or Age(\cdot, \cdot))$ gets the

[2] https://en.wikipedia.org/wiki/Impact_factor

"age" of a paper p. We can see from the first term in the equation that the younger a paper is, the sharper the trend of the changing is. The second term in the function represents the influence of the number of citations which indicates the more citations the paper has, the more important it is. Note that the exponent $\frac{1}{y(p)}$ could let the newly published papers have higher weight than the old ones.

2) **Edge-Reinforced Citation Network Construction:** Previous citation network, denoted as G, only uses boolean values 0 and 1 to denote whether there is a citation relation between two papers, which is not accurate. For instance, suppose that $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}] \in R^{N \times N}$ is the column-stochastic probability transition matrix where N is the number of vertexes of a citation network, and *OD(i)* is the out-degree of node i. If p_v cites p_u, $\mathbf{x_{i,j}} = \frac{1}{OD(i)}$, otherwise 0. It means that the Random Walk with Restart (RWR) [30] transition probability from node p_v to any of its out-neighbors p_u only depends on the out-degree of p_v (i.e., all out-neighbors are equally likely to be visited).

In view of the above-mentioned facts, we propose an Edge-Reinforced Citation Network (ERCN) which computes the elements of its transition probability matrix with the impact factor of venues and the vitality degree of two vertexes rather than the out-degree only. We show the difference of them in Figure 2. We build the ERCN based on the citation network. Specifically, we do not change the original vertexes of the citation network, but just update the weights of edges E with the above two factors. More specifically, for two vertexes p_i and p_j, we use $w(p_i, p_j)$ to denote the weight of the edge, which is calculated as follows:

$$w(p_i, p_j) = \eta(IF(v(p_i), Y) \cdot V(p_i)) + (1 - \eta)(IF(v(p_j), Y) \cdot V(p_j)) \tag{3}$$

where $\eta(0 \leq \eta \leq 1)$ is a parameter to control the importance of the two factors, and $v(p_i)$ gets the venue that p_i was published on.

3.2 Paper clustering on citation network

Traditional content-based recommendation methods need to compare the input keywords with every paper in the publication depository, which is very time-consuming. In this section, we propose a novel paper clustering algorithm which roughly puts papers of similar topics into one cluster based on the citation relationship between publications, such that we

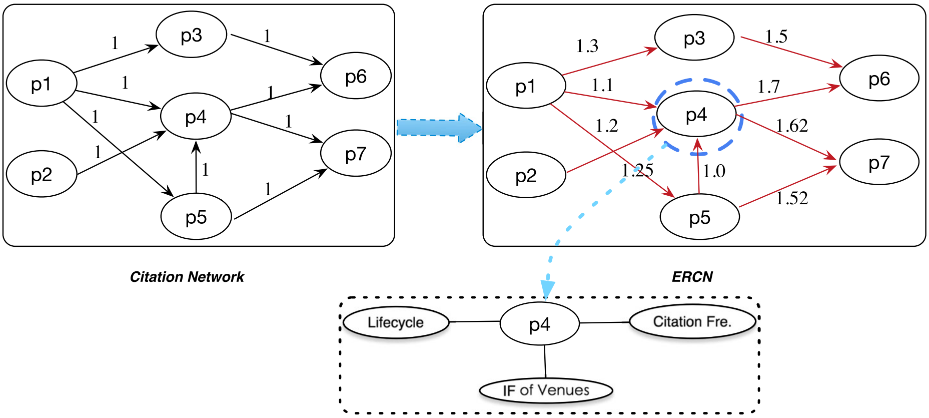


Figure 2 The transformation from citation network to ERCN

only need to do comparisons between the input keyword and the papers in relevant clusters to the input keywords.

It is nontrivial to put papers into clusters of different topics, as there are not only citation relations between papers of the same topic, but also citation relations between papers of different topics. However, we have some important observations to the citation network: (1) most of the time, each topic must have some, so called "center" paper(s), which are cited a lot by the others of the same topics; (2) papers of the same topic usually share quite a few common citation papers and common referred; (3) although it happens that a paper may cite papers of other topics, it seldom happens that many papers of one topic cite many papers of another topic. Thus, we consider to divide the citation networks into multiple sub-citation networks according to the citation relationships among papers where each sub-citation shares the same topic.

Based on the three observations above, we propose our own clustering algorithm based on the citation network: Initially, we find out a number of high-citation papers with a predefined minimum citation threshold (say 3 for instance in our experiments). Assume that there are M high-citation papers that are cited more than the predefined minimum citation threshold. We treat each of the M papers as a center for a sub-citation network such that we can get M topic clusters. After that, we can put all the left papers into the M topic clusters according to their distance to the center of each cluster.

1) **Identifying Centers for Topic Clusters:** In order to identify centers for topic clusters from these high-citation papers, we must figure out which papers should be merged into one center. Assume we have M topic clusters, but we may have a lot more high-citation papers than M in the beginning. Initially, we let each high-citation paper denote a center of a cluster, and then we keep on merging multiple centers into one center if we find out that they are very similar both on their topics and on their structures on the citation network, i.e., they are cited by the same set of papers and also cite the same set of papers.
2) **Dividing Citation Network into Clusters:** Given M centers for M topic clusters, we now consider finding the right cluster for each paper. The way we do this is to find a **Minimum Deleting-Edge-Set** . By removing this set of edges from the network, we can naturally divide the citation network into M partitions, each of which corresponds to a topic cluster. In this way, we actually transform our problem into the following optimization problem, which targets at maximizing the following formulation:

$$\max \sum_{(x_i,x_j)\in R^{M\times M}} \frac{dis(x_i,x_j)}{\sum_{x_k\in Con(x_i,x_j)} dis(x_i,x_k)+dis(x_j,x_k)+\gamma} \tag{4}$$

   where $dis(x_i,x_j)$ gets the distance of the x_i and x_j which can be measured by the topic similarity of papers. (x_i,x_j) are the combination of centers of citation network. $Con(x_i,x_j)$ is the set of nodes connecting the center nodes x_i and x_j. γ is equilibrium factor to prevent the denominator being zero.

This problem is a NP-hard one, which can be reduced from the balanced max-skip partitioning problem [33]. In the following, we employ a greedy algorithm to solve the problem, which always greedily deleting the edge connecting different centers directly or indirectly, until no more centers are connected.

We find the nodes which are shared by centers and the node set where one center can reach another center through it. The edges of them connecting the centers are the candidate deleting edges denoted as $CanEd = \{e_1, e_2, ..., e_z\}$. We estimate the closeness degree of the citation network c as follows:

$$Clo(c) = \frac{\sum_{(x_i,x_k)\in E} dis(x_i, x_k)}{\sum_{x_k \in Con(x_i,x_j)} dis(x_i, x_k) + dis(x_j, x_k) + \gamma} \tag{5}$$

where x_i and x_j are center nodes, and x_k is the node connecting center nodes. E is the set of edges of the citation network c and $Con(x_i, x_j)$ is the set of nodes connecting the node x_i and x_j. γ is equilibrium factor to prevent the denominator being zero. For each candidate edge e_i from $CanEd$, we estimate the closeness degree after deleting the edge e_i for the citation network c which generates two clusters c_1 and c_2. If the follow equations are satisfied, we will delete it.

$$\begin{cases} Clo(c) \le Clo(c_1) + Clo(c_2) \\ |Clo(c_1) - Clo(c_2)| \le \min\limits_{(x_i,x_k)\in E} dis(x_i, x_k) \end{cases} \tag{6}$$

We iteratively execute the above step until the center nodes are not connected with each other. Then for each citation network, we generate many sub-citation networks which are the so-called clusters.

Example 1 As depicted in Figure 3, we may first get 4 centers (p_1, p_6, (p_7, p_{11}) and p_{13}) with our analysis, and then find and delete edges ($< p_7, p_4 >$, $< p_7, p_6 >$, $< p_{11}, p_{14} >$, $< p_{14}, p_{13} >$) from the graph to finally get 4 clusters from the citation network.

3.3 Identifying Top-*K* recommendation results

The paper clustering algorithm above will generate a number of paper clusters, each of which has a number of papers sharing a similar topic. In this subsection, we introduce how we identify the top-K most relevant papers from these clusters to the given keywords. Assume that the number of clusters is larger than K, the top-K most relevant papers to the given keywords must be within the top-K most relevant clusters to the given keywords. Thus, we first find the top-K most relevant clusters to the input keywords based on their similarities on topics and contents, which should contain the top-K most relevant papers to the input keywords.

In the following, we first introduce how we do topic-based similarity calculation to find out the top-K most relevant clusters to the input keywords, and then present how we identify the top-K most relevant papers to the input keywords from the K clusters.

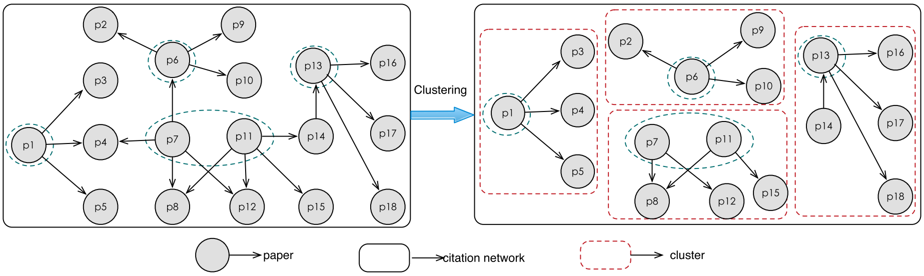


Figure 3 Paper clustering example on citation network

1) **Finding Top-K Most Relevant Clusters:** Assume there are M different topic clusters, for any cluster, we first get the each paper's topic distribution in the form of word vector, and then extract the same words from word vectors as the cluster's topic distribution. At last, we calculate the similarity between our input keywords and each cluster's word vector to find out the top-K most relevant clusters to the input keywords.

   Assume there are N distinct words $\mathbf{W} = (word_1, word_2, ..., word_N)$ in the publication depository, and for each word we calculate its TF-IDF score in this publication depository as its weight, such that we will have a normalized weight vector for all the N words as: $W =< w_1, w_2, ..., w_N >$. For each paper p, we can utilize the LDA topic model to get its topic distribution, including "document-topic" distribution and "topic-word" distribution, corresponding to the parameters θ and φ respectively as follows:

$$\theta(p,t) = \frac{Fre(t,p) + \alpha_t}{\sum_{t=1}^{M} Fre(t,p) + \alpha_t} \tag{7}$$

$$\varphi(t, word) = \frac{Fre(word,t) + \beta_w}{\sum_{i=1}^{N} Fre(word_i,t) + \beta_w} \tag{8}$$

   where $Fre(t,p)$ is the number of words belonging to a topic t in the paper p, $Fre(w,t)$ is the number of words belonging to t, and the two constants α_t and β_w are the document-topic dirichlet priori parameter and topic-word dirichlet prior parameter respectively. Note that α_t is computed with the number of topics M ($\alpha_t = \frac{50}{M}$) and β_w is usually set as 0.1. Assume cluster C_i includes L papers $P = \{p_1, p_2, ..., p_L\}$. For each paper p_j from C_i, we can gets its word weight vector $W(p) =< w_1(p), w_2(p), ..., w_n(p) >$ from its paper title and abstract, where $w_i(p) = n_i(p) * w_i$ where $n_i(p)$ denotes the number of the i-th word in paper p. Besides, we can also calculate the importance of a topic t to the paper p_j as follows:

$$w(p,t) = \frac{\sum_{word \in p} \varphi(t, word)}{\sum_{t \in \mathbf{T}} \sum_{word \in p} \varphi(t, word)} \cdot \log_2 \frac{|\mathcal{P}|}{Fre(t,p)} \cdot \frac{\theta(p,t)}{\sum_{t \in T} \theta(p,t)} \tag{9}$$

   For the given input keywords $\mathcal{KW} =< kw_1, kw_2, ..., kw_m >$, we then calculate the relevance with the word weight vector between our input keyword and the paper p_j from C_i with the equation below:

$$Rel(\mathcal{KW}, W(p)) = \frac{\sum_{i=1}^{s} \sum_{j=1}^{t} [w_i(p) \cdot w(p,t_j)]^2 \cdot Cos(\mathcal{KW}, W(p))}{\sum_{i=1}^{s} \sum_{j=1}^{t} [w_i(p) \cdot w(p,t_j) \cdot Cos(\mathcal{KW}, W)]^2} \tag{10}$$

   where $Cos(\cdot,\cdot)$ is the Cosine similarity function to compute the similarity between vectors. And then we compute the similarity of the input keywords and the cluster C_i with the sum of its papers' relevance as follows:

$$Sim(C_i, \mathcal{KW}) = \sum_{p_j \in P} Rel(\mathcal{KW}, W(p_j)) \tag{11}$$

   $Sim(w(C_i), \mathcal{KW}) = \frac{\cap(w(C_i), \mathcal{KW})}{\cup(w(C_i), \mathcal{KW})}$, where $\cap()$ and $\cup()$ get the intersection and union of word vectors respectively. Now we can build the connection between our input and the clusters with the similarity of clusters and the input keywords.

2) **Finding Top-K Ranking Papers in Each Cluster:** For each of the Top-K clusters, we employ RWR algorithm to find the top-K ranking papers in each cluster, such that we will generate K lists of sorted papers, each of which contains no more than K papers that already ranked according to their RWR scores in corresponding clusters. However, it is meaningless to compare the RWR scores of different clusters. As an alternative, we

turn to measure the topic-based similarity between each paper and the input keywords such that we will find top-K papers that are most similar to the input keywords on their topics and contents.

We first introduce how we employ the RWR algorithm to find the top-K ranking papers in each cluster. For a given node i from a cluster, the RWR proximity values from this node to other nodes are shown with following formulation.

$$\mathbf{p}_i = (1-\alpha)\mathbf{X}\mathbf{p}_{i-1} + \alpha\mathbf{e}_i \tag{12}$$

where $\mathbf{p}_i \in R^N$ is the relevance vector of node i; $\mathbf{e}_i \in R^N$ is a unit vector having $\mathbf{e}_i(j) = 1$ when $i = j$ and all other values are 0, and $\alpha \in [0, 1]$ denotes the restart probability in RWR (typically, $\alpha = 0.15$). $\mathbf{X}$ is the newly generated probability transition matrix for the cluster, where $\mathbf{x_{i,j}} = \frac{w(p_i, p_j)}{\sum_{j=1}^{N} w(p_i, p_j)}$ if there exists citation relationship, otherwise 0. We run RWR algorithm iteratively on the cluster to compute/update $\mathbf{p}_u$ with the Equation 12 until it converges or satisfies the stopping condition (such as a predefined iteration times).

3) Getting Top-K Most Relevant Papers: Given K lists of sorted papers, each of which contains no more than K papers that are already ranked according to their RWR scores in corresponding clusters as the example shown in Figure 4, we now introduce how we get top-K papers that are most similar to the input keywords on their topics and contents.

Since the way we calculate the relevance between a given paper p and our input keywords is similar to the way to calculate the similarity between a topic cluster and the input keywords, we do not use further space to describe the details here. However, if we do pairwise comparison between each paper in the K-lists of papers and the input keywords, we may need $K * K$ times of comparisons in total which is not optimal. In the following, we

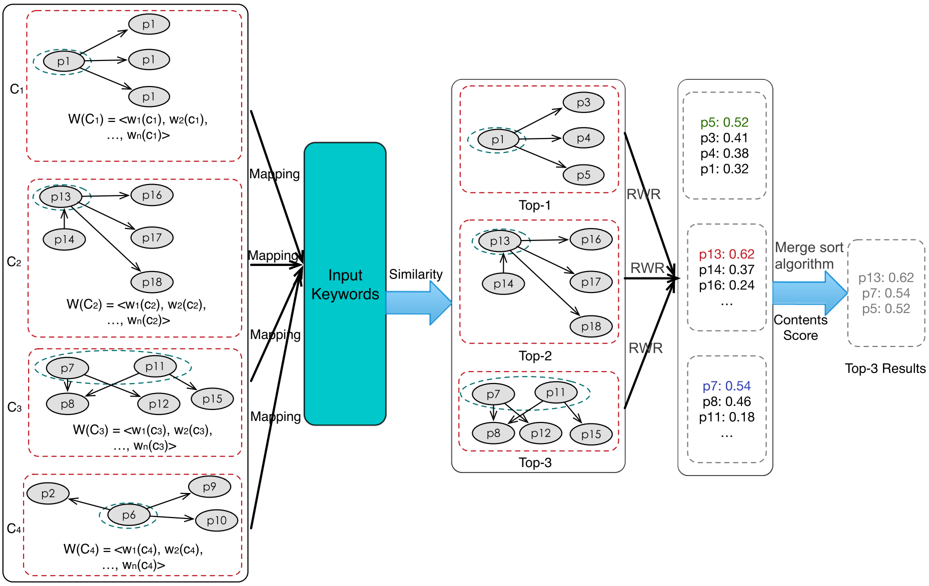


Figure 4 Getting the top-k most relevant papers to the input keywords (Let $K = 3$)

introduce an algorithm that we employ to get top-K most relevant papers from the K-list of papers efficiently.

Given the K lists of papers, each of which is sorted with their RWR scores calculated within each cluster, we only need to calculate the relevant between the first paper of each list with the keywords, which uses us K times comparisons, and the one with the highest relevant score will definitely the top-1 most relevant paper to the keywords. We then get the paper from the list that just generates the “winner” of the last round and calculate the relevance between the paper and the keywords and then again find the next “winner” of this round. We keep on repeating the two steps until we have collected all the top-K most relevant papers to the keywords.

4 Experiments

In this section, we report our experimental study results. We first describe the datasets used in our experiments. Then we introduce the experimental settings including the compared methods contrasting to our proposed method and the metrics to evaluate these methods. We next present their performance and offer the performance analysis. At last, we conduct case studies to demonstrate its effectiveness.

4.1 Data preparation

In the experiments, we use two bibliographic datasets, the DBLP dataset[3] [35] and Microsoft Academic dataset.[4]

1) *DBLP dataset.* This datasets is a subset of “DBLP V1” provided by Tsinghua University for their ArnetMiner academic search engine [35]. We preprocessed this dataset with the following rules: 1) If the number of citations of a paper is less than 2, we delete it. 2) If there exists missing information of a paper, such as title, venue, abstract etc. , we eliminate it. After the preprocessing, we get 34104 papers and 78841 citations.
2) *Microsoft Academic dataset.* This dataset, MAS, contains many research fields. Here, we selected the Computer Science domain and queried the engine for the 300 conferences which published at least one paper. Then, for each conference, we queried MAS for the last 200 published papers and discarded those where there exists missing information. We extract abstracts of papers from DBLP to combine the information of MAS. At the end, we collected a dataset with 101, 205 papers and 190, 146 citations partitioned almost uniformly among 300 conferences.

Here, we construct the basic citation networks based on the citation relationship between papers for the above two datasets. More specifically, if a paper p_i cites another paper p_j, then there is a directed edge from p_j to p_i. For the construction of ground truth of our datasets, we directly treat the real citation relationship from the citation networks as the known-knowledge, which is consistent with the facts. This means that we select any Newly published papers from the citation network and validate whether they are included in the result sets achieved with our proposed method. For a given input corresponding to a paper p, if it is included in the results, this indicates our recommendation for the input is useful.

[3] http://arnetminer.org/DBLP_Citation

[4] http://academic.research.microsoft.com

4.2 Experimental settings

We provided details on the experimental settings for conducting evaluations on all the methods.

4.2.1 Compared methods

We compared our proposed method with its variation which considered only one or two aspects contrasting to our method which utilizes the compound information of papers.

- **Contents-based Paper Recommendation (CPR).** This method only uses the contents of papers to calculate the relevance of our input. It first gets the topic distribution of papers with LDA model and then acquires the word vector presenting the key information of papers. At last, it calculates the relevance between the papers from publication depository and our input keywords with similarity function.
- **RWR Paper Recommendation (RWRPR).** This approach uses the citation networks to recommend papers. For every citation network, we map our input keywords into possibly related citation networks based on the topic distribution. Finally, we generate a virtual nodes in citation networks representing a paper which connects with other papers having same weight 1.0 and then run the RWR algorithm and at last calculate the relevance.
- **ClusCite approach** [31]**.** ClusCite is a cluster-based citation recommendation framework to put the softly clustered into interest groups using the multiple types of relationship of the network. It considered the context of heterogenous bibliographic network which believed each group has its own model for paper authority and relevance.
- **PubTeller No Cluster (PTNC).** PTNC is a method which employs the combination of contents-based method and citation network based method. Note that the contents-based method does not use the citation network clustering. And the citation network based method utilizes the Edge-reinforced citation network (ECRN) which considers the impact of venues and vitality degree of papers.
- **PubTeller with Cluster (PTC).** This method is our proposed method which not only leverages the citation network clustering based contents recommendation but also ERCN based recommendation to get the related papers for our input.

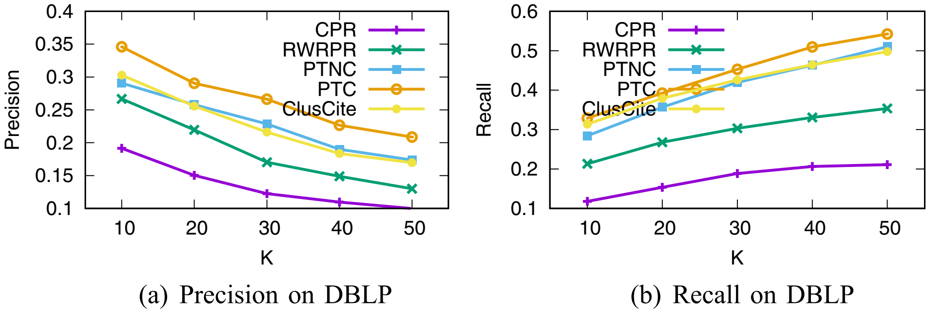
4.2.2 Evaluation metrics

We utilize Precision and Recall at position K (P@K and R@K) as the evaluation metrics. Recall@K is the percentage of original papers that appear in the top-K recommended list. A high recall with a lower K indicates a better paper recommendation system. Precision@K was used to measure the effectiveness of the recommendation system by checking whether the original papers were ranked high for the query manuscript.

4.3 Performance comparison

We now compare the proposed recommendation model (PubTeller) with other baseline in terms of contents-based recommendation performance and citation recommendation performance.

As shown in Figure 5, the contents-based method has the lowest precision comparison to other methods when K varies from 10 to 50. This shows that only relying on the



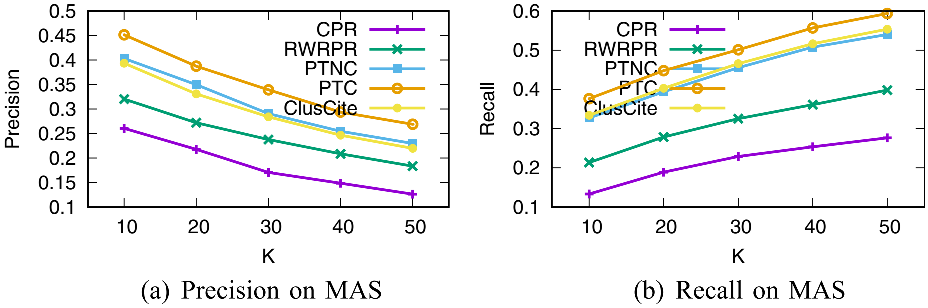
(a) Precision on DBLP (b) Recall on DBLP

Figure 5 Performance comparison on DBLP

content can not precisely reflect the relevance due to the similar topic features. The RWR method works better than content-based method since it considers the citation relationship of papers which employs the classical PageRank algorithm to compute the relevance. It can capture more features of papers. Our proposed baseline algorithm PTNC gets the similar performance with ClusCite approach. This method does not utilize the paper clustering algorithm but uses the combination of proposed edge-reinforced citation network and contents of papers to calculate the relevance. The traditional RWR considers the relationship of papers only with the in-degree and out-degree of papers while ERCN not only leverages the impact factor of venues but also takes into consideration the vitality degree of papers. With these two features, we can acquire more accurate information of papers. Our proposed PTC approach gets the best performance than other methods. It not only utilizes the ERCN to run RWR algorithm but also uses the paper clustering algorithm to put papers with similar topic features together such that the accuracy of paper recommendation can be improved. Besides, we also employ contents of papers to increase the accuracy further. As we can see in Figure 5, our PTC algorithm improves precision on average 3.2% and recall on average 4.1% than ClusCite. In Figure 6, we can see that our PTC also outperforms other methods on MAS dataset.

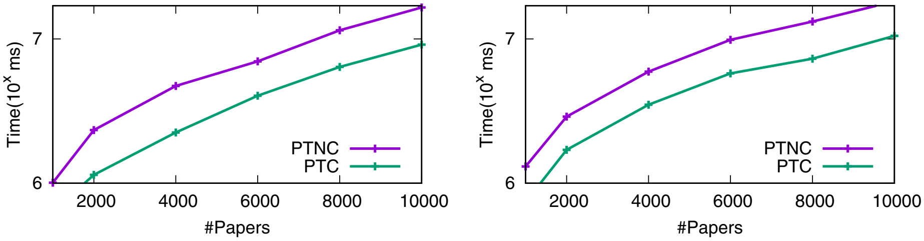
4.4 Efficiency comparison

In this section, we compare the efficiency of our proposed methods with clustering (PTC) and without clustering (PTNC). As we can see in Figure 7a, with the number of papers



(a) Precision on MAS (b) Recall on MAS

Figure 6 Performance comparison on MAS



(a) Efficiency on DBLP

(b) Efficiency on MAS

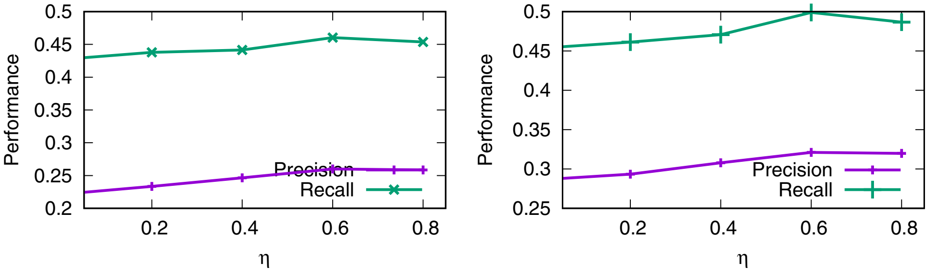
Figure 7 Efficiency comparison on two datasets

increasing, the time cost of paper recommendation increases for PTC and PTNC. However, the efficiency of PTC is much better than PTNC. The former saves about 40% time cost compared to the later on the DBLP dataset. The reason is that PTC decreases the comparison times a lot by paper clustering. When calculating the relevance of papers, we just compare papers from the related clusters rather than the citation network it belongs to. As illustrated in Figure 7b, PTC also gets the better efficiency than PTNC on the MAS dataset.

4.5 Parameter study

We study the impact of the parameter η for the precision and recall on two datasets in this section. The parameter η is to control the weights of reference node and referenced node for the important of IF value and vitality degree. We employ the linear weighted method to assign different attentions on above two parts by setting different values of η. In order to compare the effect of the parameter η conveniently, here we select the top K (K=30 in our experiments) recommended papers to compute the precision and recall. Surely, we can also set K with different values.

As shown in Figure 8a and b, the value of η has the influence on the performance of paper recommendation, i.e. precision and recall. When increasing η from 0 to 0.8, the precision and recall first increase slowly before $\eta = 0.2$ and after that they increase slightly quicker than before and finally the trends tend to vary slowly after $\eta = 0.6$ on the two datasets. The reason is that when η is small, the referenced node only has a small effect, but when it becomes larger, the referenced node gets more attentions on the Edge-reinforced Citation Networks. But after reaching at some point, this influence diminishes. In addition,



(a) η Impact on DBLP

(b) η Impact on MAS

Figure 8 Parameter influence on precision and recall on two datasets

it indicates that the performances are effected more by the referenced node than the reference node. With our experiments, we find that when η is 0.65, the performance gets the best results.

5 Related work

So far, plenty of work has been done in paper recommendation, which can be roughly divided into three branches, i.e., (1) citation network-based methods [12, 13, 27], (2) content-based methods [3, 14, 26] (3) co-author network-based methods [21, 23, 29] and hybrid methods [24, 37, 40].

5.1 Citation network-based recommendation methods

Methods of this kind aim at mining the citation relationship between items, and then employing Random Walk on the citation network to find the most relevant items for inputs. Yicong Liang et al. proposed a method [19] to address the problem by incorporating various citation relations for a proper set of papers. Their method used Local Relation Strength to measure the dependency between cited and citing papers and Global Relation Strength model to capture the relevance between two papers in the whole citation graph. However, the approach will be inefficient if we want relevant papers whose topics have strong interconnection with the topic of a given paper. Huang et al. proposed a Citation Semantic Link Network (C-SLN) to describe the semantic information on citation networks [10]. This method employed NLP tools to build C-SLN and then calculated the importance of references. They assumed that if a reference appeared many times in the main part of an article, it should be given a higher importance. But the process of extracting the occurrence and position of each reference can be very time-consuming. Similar to citation network, Jiemin Chen et al. put forward a community-based scholar commendation model [5], which needs constructing research-fields-based graphs firstly, detecting communities in the graphs and then making scholar recommendation by calculating friendship scores. The experimental results demonstrated that the approach outperforms the content-based user recommendation method . But they needed extra times to construct graphs and it gave up more useful research information from scholars.

5.2 Content-based recommendation methods

Contents-based Recommendation intend to leverage the contents of papers to find the relevance between them. That is, they mine the information from contents with text analysis tools, and then calculate the similarities between the inputs keywords and candidates from paper set, finally recommend the most relevant papers. Ekstrand et al. proposed several methods for augmenting Collaborative Filtering [11, 20] and content-based filtering algorithms with measures of the influence of a paper within the Web of citations so as to recommend suitable papers to users [6]. However, this method needed to know topics of papers in advanced which users want to read. When we do not have these information, it cannot work well. Besides, a LDA-based method was proposed to recommendation papers [1] which employed topic models to build the users' profile based on their published papers and language model to get the topics' distribution of papers, leveraged their similarity to present the ranking scores to recommend papers. Some other approaches were also proposed which used latent topic models to recommend papers by modeling citation links jointly [28, 34], such as Link-PLSA-LDA and TopicSim.

5.3 Co-author network-based recommendation methods

This type of methods recommend relevant papers to users based on the co-author relationship. They assume that if a user is interested in papers written by an author, he/she is also likely to prefer to read papers written by the author's co-authors. Jing Li et al. did some research on recommending based on co-authorship, where they proposed a random walk model [18] using three specific academic network metrics including coauthor order, collaboration time points and frequency of collaboration to improve the recommendation quality and accuracy. The approach only count on three academic metrics while many other features exist, such as citation relationship. And there are many other works using RWR for paper recommendation [15, 17, 25] which are the improved RWR to recommend papers accurately.

5.4 Hybrid recommendation methods

In addition, some hybrid recommendation methods are also proposed which combine some of the above three types of methods with traditional recommendation approaches used by recommendation system, or integrate some of them to recommend the most relevant papers. For instance, Torres et al. proposed a combination of Content-based Filtering and Collaborative Filtering algorithm to recommend research papers to users [36]. Yang et al. proposed a joint multi-relational model that can exploit the latent correlation among author-paper-citation relation, author-author collaboration relation, author-paper- venue relation and paper-paper-citation relation [39]. Lu et al. proposed an academic resource recommendation method which integrated the Advanced Hyperlink Induced Topic Search(AHITS) algorithm and User-Paper-Topic(UPT) model [22]. AHITS was used to evaluate the quality and authority of academic resources while UPT was employed to model user research interest based on constructing a tripartite graph.

In this paper, we propose an approach using the compound information of papers to recommend papers, such as contents , citation networks, the citation years of papers and the impact factors of venues and so forth. A citation network cluster method is proposed to decrease the comparison times while using the contents to calculate the relevance and an edge-reinforced citation network is put forward to compute the ranking score which considers the citation years of papers and the impact factor of venues. With this approach, we can greatly improve the efficiency and effectiveness of paper recommendation.

6 Conclusions and future work

We work on employing compound information of papers to recommend papers in this paper. The traditional methods face the problem of low accuracy. To solve the problem, we propose a novel hybrid publication recommendation algorithm using compound information, PubTeller, which not only considers the contents but also the citation network and vitality degree of papers. Considering the low efficiency of traditional paper recommendation, we propose a citation network clustering method to decrease the comparison times for our input. Besides, we construct an edge-reinforced citation network to capture the relevance of papers more accurately than traditional citation network. For a given tittle, PubTeller recommends top-K relevant papers. Extensive experimental results based on several data collections demonstrate that our proposed PTC can effectively and accurately recommend the most relevant papers from immense datasets which meets the interests of users and thus help improve the accuracy on precision about 4% and recall about 4.5%. Our PTC algorithm saves the time cost on average 40% for the PTNC algorithm. As a future work, so as to improve the precision of our proposed method, we would like to extend our work with crowdsourcing.

Acknowledgements This research is partially supported by National Natural Science Foundation of China (No. 61632016, 61572336, 61572335, 61772356), the Natural Science Research Project of Jiangsu Higher Education Institution (No. 17KJA520003, 18KJA520010), the Open Program of Neusoft Corporation (No. SKLSAOP1801), and King Abdullah University of Science and Technology (KAUST) under award number FCC/1/1976-19-01.

References

1. Amami, M., Pasi, G., Stella, F., Faiz, R.: An Lda-Based approach to scientific paper recommendation. In: International Conference on Applications of Natural Language to Information Systems, pp. 200–210. Springer (2016)
2. Beel, J., Langer, S., Genzmehr, M., Gipp, B., Breitinger, C., Nürnberger, A.: Research paper recommender system evaluation: a quantitative literature survey. In: Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, pp. 15–22. ACM (2013)
3. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Commun. ACM **35**(12), 29–38 (1992)
4. Cazella, S.C., Alvares, L.O.C.: An architecture based on multi-agent system and data mining for recommending research papers and researchers. In: Eighteenth International Conference on Software Engineering & Knowledge Engineering, pp. 67–72 (2006)
5. Chen, J., Tang, Y., Li, J., Mao, C., Xiao, J.: Community-based scholar recommendation modeling in academic social network sites. In: International Conference on Web Information Systems Engineering, pp. 325–334. Springer (2013)
6. Ekstrand, M.D., Kannan, P., Stemper, J.A., Butler, J.T., Konstan, J.A., Riedl, J.T.: Automatically building research reading lists. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 159–166. ACM (2010)
7. Gori, M., systems, A.Pucci.: Research paper recommender a random-walk based approach. In: 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06), pp 778–781. IEEE (2006)
8. He, Q., Pei, J., Kifer, D., Mitra, P., Giles, L.: Context-aware citation recommendation. In: Proceedings of the 19th international conference on World wide Web, pp. 421–430. ACM (2010)
9. Huang, W., Wu, Z., Chen, L., Mitra, P., Giles, C.L.: A neural probabilistic model for context based citation recommendation. In: AAAI, pp. 2404–2410 (2015)
10. Huang, Z., Qiu, Y.: A multiple-perspective approach to constructing and aggregating citation semantic link network. Futur. Gener. Comput. Syst. **26**(3), 400–407 (2010)
11. Huang, Z., Zeng, D., Chen, H.: A comparison of collaborative-filtering recommendation algorithms for e-commerce. Intelligent Systems IEEE **22**(5), 68–78 (2007)
12. Küçüktunç, O., Saule, E., Kaya, K., Çatalyürek, Ü..V.: Direction awareness in citation recommendation. Wien. Med. Wochenschr. **123**(9), 148–149 (2012)
13. Küçüktunç, O., Saule, E., Kaya, K., Çatalyürek, Ü.V.: Recommendation on academic networks using direction aware citation analysis. arXiv:1205.1143 (2012)
14. Lang, K.: Newsweeder: Learning to filter news (1995)
15. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Mach. Learn. **81**(1), 53–67 (2010)
16. Lee, J., Lee, K., Kim, J.G.: Personalized academic research paper recommendation system. Computer Science (2013)
17. Li, J., Willett, P.: Articlerank: a pagerank-based alternative to numbers of citations for analysing citation networks. In: Aslib Proceedings, vol. 61, pp. 605–618. Emerald Group Publishing Limited (2009)

18. Li, J., Xia, F., Wang, W., Chen, Z., Asabere, N.Y., Jiang, H.: Acrec: a co-authorship based random walk model for academic collaboration recommendation. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 1209–1214. ACM (2014)
19. Liang, Y., Li, Q., Qian, T.: Finding relevant papers based on citation relations. In: International Conference on Web-Age Information Management, pp. 403–414. Springer (2011)
20. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
21. Lopes, G.R., Moro, M.M., Wives, L.K., De Oliveira, J.P.M.: Collaboration recommendation on academic social networks. In: International Conference on Conceptual Modeling, pp. 190–199. Springer (2010)
22. Lu, M., Wei, X., Gao, J., Shi, Y.: Ahits-upt: A high quality academic resources recommendation method. In: IEEE International Conference on Smart City/Socialcom/Sustaincom, pp. 507–512 (2015)
23. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Forth International Conference on Web Search and Web Data Mining, WSDM 2011, pp. 287–296. Hong Kong (2011)
24. Ma, K., Lu, T., Abraham, A.: Hybrid Parallel Approach for Personalized Literature Recommendation System. In: International Conference on Computational Aspects of Social Networks, pp. 31–36 (2014)
25. Ma, N., Guan, J., Zhao, Y.: Bringing pagerank to the citation analysis. Inf. Process. Manag. **44**(2), 800–810 (2008)
26. Massa, P., Avesani, P.: Trust-Aware Collaborative Filtering for Recommender Systems. Springer, Berlin (2004)
27. Mcnee, S.M., Albert, I., Dan, C., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Cscw02, P, pp. 116–125 (2003)
28. Nallapati, R.M., Ahmed, A., Xing, E.P., Cohen, W.W.: Joint latent topic models for text and citations. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 542–550. ACM (2008)
29. Newman, M.E.: Scientific collaboration networks. i. network construction and fundamental results. Phys. Rev. E **64**(64), 016131 (2001)
30. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the Web (1999)
31. Ren, X., Liu, J., Yu, X., Khandelwal, U., Gu, Q., Wang, L., Han, J.: Cluscite: Effective citation recommendation by information network-based clustering. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pp. 821–830. ACM (2014)
32. Salton, G.: Associative document retrieval techniques using bibliographic information. J. ACM (JACM) **10**(4), 440–457 (1963)
33. Sun, L., Franklin, M.J., Krishnan, S., Xin, R.S.: Fine-grained partitioning for aggressive data skipping. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 1115–1126. ACM (2014)
34. Tang, J., Zhang, J.: A Discriminative Approach to Topic-Based Citation Recommendation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 572–579. Springer (2009)
35. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 990–998. ACM (2008)
36. Torres, R., McNee, S.M., Abel, M., Konstan, J.A., Riedl, J.: Enhancing digital libraries with techlens+. In: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries, pp. 228–236. ACM (2004)
37. Wang, Q., Li, W., Zhang, X., Lu, S.: Academic paper recommendation based on community detection in citation-collaboration networks (2016)
38. Wang, Y., Zhai, E., Hu, J., Claper, Z.Chen.: Recommend classical papers to beginners. In: Seventh International Conference on Fuzzy Systems and Knowledge Discovery, pp. 2777–2781 (2010)
39. Yang, Z., Yin, D., Davison, B.D.: Recommendation in academia a joint multi-relational model. Ieee/Acm International Conference on Advances in Social Networks Analysis and Mining, pp. 566–571 (2014)
40. Zhang, P.Y., Du, Y.J., Wang, C.: A hybrid method based on hits for literature recommendation. Appl. Mech. Mater. **55-57**, 1636–1641 (2011)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Qiang Yang[1,2] · Zhixu Li[1,3] · An Liu[1] · Guanfeng Liu[1,4] · Lei Zhao[1] · Xiangliang Zhang[2] · Min Zhang[1] · Xiaofang Zhou[1,5]

Qiang Yang
qiangyanghm@hotmail.com

An Liu
anliu@suda.edu.cn

Guanfeng Liu
guanfeng.liu@mq.edu.au

Lei Zhao
leizhao@suda.edu.cn

Xiangliang Zhang
xiangliang.zhang@kaust.edu.sa

Min Zhang
minzhang@suda.edu.cn

Xiaofang Zhou
zxf@uq.edu.au

[1] Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou, China

[2] King Abdullah University of Science and Technology, Jeddah, Saudi Arabia

[3] IFLYTEK Research, Suzhou, China

[4] Department of Computing, Macquarie University, Sydney, Australia

[5] The University of Queensland, Brisbane, Australia