This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

VOPRec: Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation

Xiangjie Kong, *Senior Member, IEEE*, Mengyi Mao, Wei Wang, Jiaying Liu and Bo Xu

Abstract—Finding relevant papers is a non-trivial problem for scholars due to the tremendous amount of academic information in the era of scholarly big data. Scientific paper recommendation systems have been developed to solve such problem by recommending relevant papers to scholars. However, previous paper recommendations calculate paper similarity based on hand-engineered features which are inflexible. To address this problem, we develop a scientific paper recommendation system, namely VOPRec, by vector representation learning of paper in citation networks. VOPRec takes advantages of recent research in both text and network representation learning for unsupervised feature design. In VOPRec, the text information is represented with word embedding to find papers of similar research interest. Then, the structural identity is converted into vectors to find papers of similar network topology. After bridging text information and structural identity with the citation network, vector representation of paper can be learned with network embedding. Finally, top-Q recommendation list is generated based on the similarity calculated with paper vectors. Through the APS data set, we show that VOPRec outperforms state-of-the-art paper recommendation baselines measured by precision, recall, F1, and NDCG.

Index Terms—Network Representation Learning, Paper Recommendation, Graph Learning, Scholarly Big Data.

1 Introduction

Recent years, as more and more scientific papers are produced, the rapid growing scholarly data have been witnessed, and the term "Scholarly Big Data" is coined which contains millions of authors, papers, citations, tables, as well as massive scale related data such as citation networks [1], [2]. In the academic society, the rapid rise of scholarly big data results in the academic information overload challenges. When doing research, scholars have to search, read, and analyze many scientific papers in their research filed to find previous related works that might be insightful for starting a specific research. However, due to the increasing scale of publications, it is a non-trivial problem for scholars to find relevant papers based on bibliographical search.

Scientific paper recommendation systems have been extensively developed in various digital libraries and academic search engines, for example, Springer Nature and Elsevier. Such recommendation services aim to solve the information overload problem in scholarly big data by suggesting a list of relevant papers or venues to scholars. For example, the Digital Research Assistant service by Springer Nature tries to suggest relevant papers based on your individual research interests by analysing the last 100 papers you have read across nature.com and SpringerLink.

This work was partially supported by the National Natural Science Foundation of China (61502071), the Natural Science Foundation of Liaoning Province, China (201602154), Fundamental Research Funds for the Central Universities(DUT16RC(4)63).

X. Kong, M. Mao, W. Wang, J. Liu and B. Xu are with Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: xjkong@ieee.org; fqxxmmy@163.com; ehome.wang@outlook.com; jiaying_liu@outlook.com; boxu@dlut.edu.cn).

The corresponding author of this paper is Bo Xu, (e-mail: boxu@dlut.edu.cn).

Scientific paper (or literature) recommendation has also been an interesting research topic in academia. According to statistics in the survey paper by Beel et al. [3], there are more than 200 research articles published about research paper recommendation systems in the last 16 years. Scientific paper recommendation systems can be divided into three categories based on the adopted techniques including, content-based filtering, collaborative filtering, and graph-based recommendation. However, these approaches focus either on the research topic extracted from paper content [4], [5] or network structure extracted from citation network [6], [7]. Although some hybrid recommendation approaches have been proposed [8], the adopted structural topologies are still hand-engineered features which are inflexible and designing such features may be time-consuming process.

The main goal of paper recommendation is to calculate the paper similarity. Recently, there is an increasing trend in learning network embeddings to get low dimensional latent vectors with unsupervised feature learning approach using neural network [9], [10], [11]. Many works have shown the effectiveness of such approach [9], [12], [13], [14], [15]. Besides, some works [16], [17], [18], [19] express the idea of mixing content and topological structure. Inspired by the idea of network representation learning, we propose VOPRec which tries to learn **V**ector representation **O**f **P**apers in citation network for **Rec**ommendation considering both content information and network topologies.

The core idea of VOPRec is to represent papers with vector in the citation network so that the similarity between papers can be calculated with the vectors. Figure 1 shows the framework of VOPRec. Specifically, VOPRec first obtains the textual vectors of papers based on paper content, and paper structural vectors from structural identify. Then, the ci-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

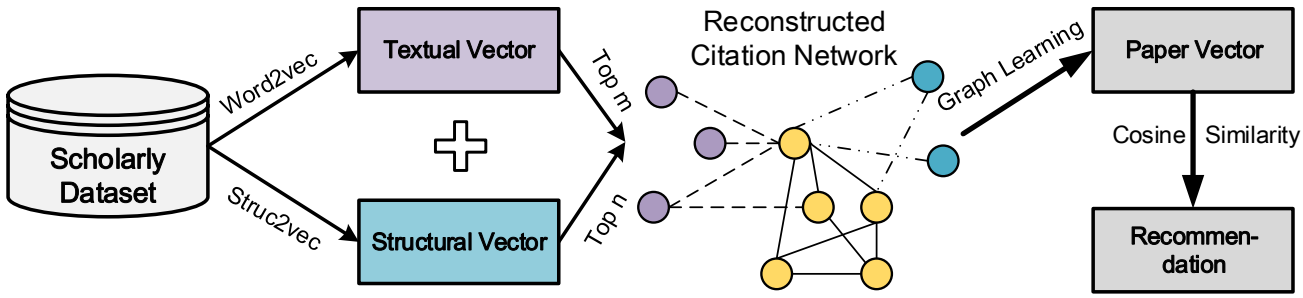


Fig. 1. Framework of VOPRec. It contains three main steps: (1) Represent papers with vectors based on context and structure; (2) Reconstruct the citation network with textual and structural vectors, where between the yellow nodes represent the citation relationships, purple nodes represent the most similar context papers and blue nodes represent the most similar structure papers; (3) VOPRec can learn paper vectors from this graph and measure similarity for paper recommendation.

tation network is reconstructed by connecting the m-nearest text-based neighbours and n-nearest structure-based neighbours. Finally, we can learn the paper vector based on the reconstructed citation network with network representation techniques. Thus, the paper similarity for recommendation can be gained by calculating the cosine similarity between paper vectors. We test the performance of VOPRec based on a real-world data set APS (American Physical Society) by comparison with content-based, collaborative filtering-based, and graph-based baseline methods. Experimental results show that VOPRec outperforms baselines in terms of precision, recall, F1, and NDCG, which demonstrates the effectiveness of VOPRec.

To the best of our knowledge, this is the first work that considers both text information and structural identity for citation network representation learning to recommend scientific papers. The main contributions of this paper can be summarized as follows:

- We present how to learn text representation of papers by combining document embedding and word embedding, which provides insight for assessing the context and topic similarity between papers.
- We present how to learn structure representation of papers by generating context of each paper from multilayer citation network, which can be used to measure the structural similarity between papers.
- We propose VOPRec, which learns vector representations of papers in the weighted citation network considering both text information and structural identity to help generate better paper recommendation results.
- We conduct extensive experiments with a real-world data set APS to evaluate our proposed method by comparing with content-based, collaborative filtering-based, and network representation learning-based baseline methods.

The rest of the paper is organised as follows: Section 2 provides the overview of the paper representation learning problems. The details of VOPRec are introduced in Section 3. The experimental results are analyzed in Section 4. We review the related work in Section 5 and Section 6 concludes this paper.

2 Overview

In this section, we first give a general overview to the paper representation learning problems. Then we introduce matrix factorization.

2.1 Problem Formulation

A citation network is very important for paper recommendation. Given a citation network as a graph $G = (V, E)$, each edge $e \in E$ denotes the citation relationship between paper v_i and paper v_j. In this citation network, we do not distinguish between the incoming and outgoing links so the graph G is undirected and $e_{i,j} = e_{j,i}$. Both cited papers and citing papers are included to help recommending papers for target papers. Based on the graph G, our task is to learn the dimensional latent representation $X \in \mathbb{R}^{|V| \times k}$, where $k \ll |V|$. In order to find the context-semantic and the network-structural relationships among these papers, a new graph $G' = \{G, E', W\}$ is built where E' is the set of new edges and $w_{i,j} \in W$ represents the weight of link $e_{i,j}$. The new graph G' is introduced in Section 3.3 in detail. The output of this task is lower dimensional matrix x_{v_i} with the k dimensions to represent the paper v_i. As a paper representation, x_{v_i} can alleviate the sparsity of network G. Moreover, we can regard x_{v_i} as the features of every paper v_i. Then, we calculate the similarity scores of every pair of papers based on the learned paper representations and choose the top-Q similar papers to recommend for target papers. The main symbols used in this paper is shown in Table 1.

2.2 Matrix Factorization

According to Yang's work [13], the network representation learning is derived from Matrix Factorization. What follows is a brief introduction of matrix factorization. In the traditional recommendation systems, since every user and every

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

TABLE 1
Main symbols in this paper

Symbol	Definition
k	Dimension of vector
m	Number of nearest text-based neighbours
n	Number of nearest structure-based neighbours
win	Training window
Q	Length of recommendation list
O	Ratio of test set

item have their own characteristic vectors, the basic idea of Matrix Factorization is to decompose the user-property matrix and feature-object matrix from the score matrix. The advantages of Matrix Factorization is to reduce the dimensions of the matrix and build the user's preferences and the characteristics of each item at the same time. Figure 2 shows the main idea of Matrix Factorization.

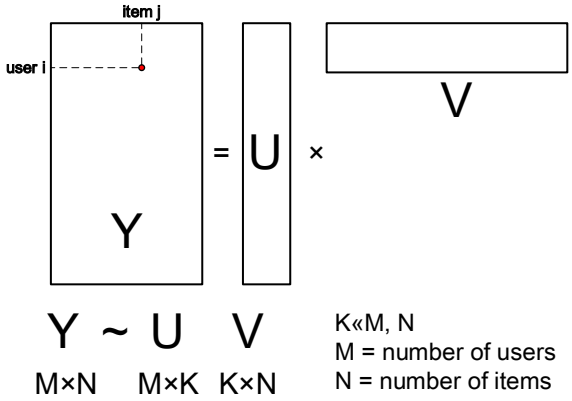


Fig. 2. Main idea of Matrix Factorization: decompose a high dimensional matrix into two low dimensional matrices.

Basic Matrix Factorization is shown in Equation 1:

$$\min_{U,V} \|Y - UV^{\mathrm{T}}\| + L(U,V), \tag{1}$$

where $L(U,V)$ is the regularization for bias-variance trade-off [20]. In the network representation, a paper is embedded into a k-dimensional vector $r_v \in \mathbb{R}^k$ and a context vertex $v \in V_P$ is represented into a k-dimensional vector $c_v \in \mathbb{R}^k$. So the user i in Figure 2 is replaced by $r_{v_i} \in W$ and U is replaced by a $k \times |V_P|$ matrix W containing relationships between papers. Similarly, the item j in Figure 2 can be replaced by $c_{v_j} \in H$ and V is replaced by a $k \times |V_P|$ matrix H containing the text information of papers. Yang et. al [13] figured out the closed form matrix M that can replace the Y in Equation 1, where $M = W^{\mathrm{T}}H$. Each entry in M in Skip-Gram with Negative Sampling is

$$M_{ij} = \log \frac{N(v_i, c_j) \cdot |D|}{N(v_i) \cdot N(c_j)} - \log a, \tag{2}$$

where D is the random walk sequences, $N(v) = \sum_{c' \in V_C} N(v, c')$ and $N(c) = \sum_{v' \in V} N(v', c)$. $N(v, c)$ denotes the number of times (v, c) appears in D, and a is

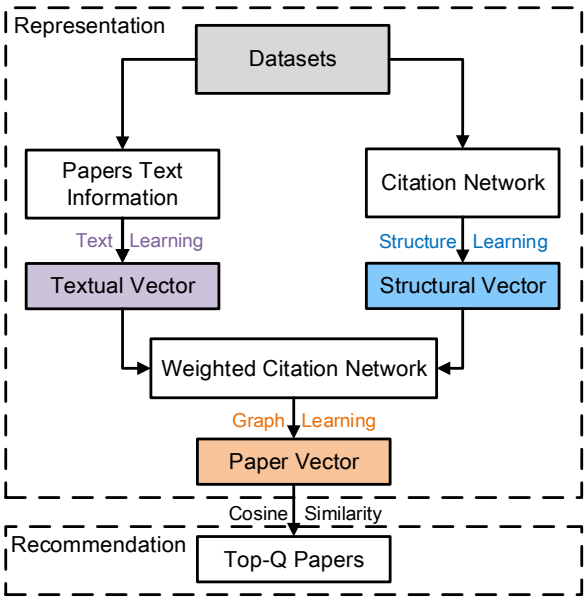


Fig. 3. The flow of VOPRec. The first part is to represent papers with vectors and the second part is to recommend papers by calculating the cosine similarity of paper vectors.

the number of negative samples for each word-context pair (v, c). The same as the matrix M in Skip-Gram is

$$M_{ij} = \log \frac{N(v_i, c_j)}{N(v_i)}. \tag{3}$$

In the network representation, we have

$$\frac{N(v_i, c_j)}{N(v_i)} = \frac{[e_i(A + A^2 + \cdots + A^t)]_j}{t}, \tag{4}$$

where A is the transition matrix in random walk, e_i is the initial state of random walk, and t represents t steps in random walk. Thus, $M_{ij} = \log N(v_i, c_j)/N(v_i)$ denotes the average probability that node i randomly walks to node j in t steps.

3 The Framework of VOPRec

In this section, we propose a framework, VOPRec, which recommends papers for target papers through learning context representations from paper text information and node representations in citation network. The complete algorithm of VOPRec is described in Algorithm 1. The flow of framework, which is shown in Figure 3, is composed of four main steps, as follows:

- **Text Learning**: Combine document embeddings and word embeddings to find a textual representation for all papers. This provides more information to assess context similarity between papers.
- **Structure Learning**: Generate context of each paper from multilayer citation network to learn structure representation. Not only the structural similarity in every layer but also the structural similarity between layers are considered.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

Algorithm 1: VOPRec

Input: Word Matrix W_T; Citation Network $G = (V, E)$; text-based neighbours m; structure-based neighbours n; embedding size k; window size $|win|$; walks per node r; walk length wl; Recommendation List length Q

Output: Recommendation List RL

```
WR =Doc2vec(W_T);
SR =Struc2vec(G);
for i = 1; i ≤ row(WR); i++ do
    get most m similar papers;
for j = 1; j ≤ row(SR); j++ do
    get most n similar papers;
Build the weighted citation network G' = (V, E ⋃ E_1 ⋃ E_2, W) with m and n neighbours for each paper;
for (i, j) ∈ G' do
    w_{i,j} = weight(i, j);
for k = 1; k ≤ r; k++ do
    for nodes u ∈ V do
        walk[u] =neighbour(u, G', wl);
return walk;
P =Skip-Gram(walk);
for t = 1; t ≤ row(P); t++ do
    RL[t] =most Q similar papers;
return RL;
```

- **Bridging Weighted-Citation Graph with Text and Structure**: Bridge the m-nearest text-based neighbours and n-nearest structure-based neighbours with the citation network, and give different weights for each edge to build the weighted graph based on text information, structure information, and citations.
- **Learning from Graph**: Add the paper paths traversed by random walk over the weighted network into the neighbour function and apply Skip-Gram model to learn each paper's latent representation. Then, these trained vectors can be used to measure the similarity for paper recommendation.

3.1 Text Learning

The first step is to acquire the textual vector for every paper $v_i \in G$. The approach for learning textual representation is inspired by the method of Paper2vec [21]. The text information of papers can be composed of the papers' titles, abstracts or the main body of the full papers. We build up a corpus of these text information of papers as a set $T = \{t_1, t_2, \cdots, t_V\}$. We learn the word vector w_i and document vector d_j based on Doc2vec [22], where each paper's information is regarded as a document. Inspired by Doc2vec, every paper can be mapped to a unique vector which is represented by a column in matrix P, and every word can be also mapped to a unique vector represented by a column in matrix WD. Figure 4 shows the process for learning textual paper vectors.

The paper vector could be regarded as another word vector actually, which plays the role in filling the vacancy

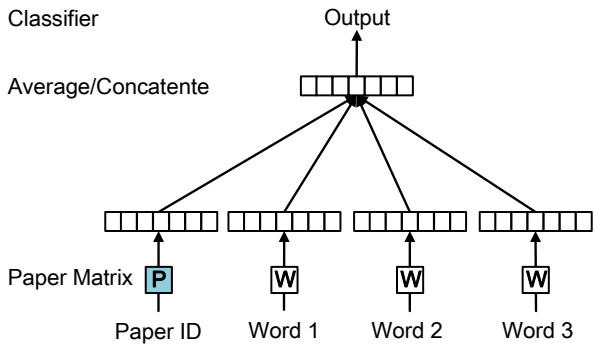


Fig. 4. Paper vector based on text.

information. For a given sequence of the training words $w_1, w_2, w_3, \cdots, w_T$, we want to fix these training words in a predefined window win which slides throughout our corpus. In other words, we train every word in the context of length of win to predict the possibility of every word in this context given the paper vector and word vector itself. Then the main work is to maximize the average log probability,

$$\frac{1}{|win|} \sum_{t=l}^{|win|-l} \log p(w_t | w_{t-l}, \cdots, w_{t+l}, d_j). \tag{5}$$

Next, the prediction task is assigned to the multiclass classifier, we use hierarchical softmax based on the binary Huffman tree, where short codes are assigned to frequent words,

$$p(w_t | w_{t-l}, \cdots, w_{t+l}, d_j) = \frac{\exp(w_t^{\mathrm{T}} \cdot (w_{t-l}, \cdots, w_{t+l}) + w_t^{\mathrm{T}} d_j)}{\sum_{i=1}^{|win|} \exp(w_i^{\mathrm{T}} \cdot (w_{t-l}, \cdots, w_{t+l}) + w_i^{\mathrm{T}} d_j)}. \tag{6}$$

The objective function can be optimized by stochastic gradient descent algorithm.

Finally, we can get the normalized log-probability for each output paper i, which is computed as

$$y = b + Uh(w_{t-l}, \cdots, w_{t+l}; WD, DC), \tag{7}$$

where U and b are the softmax parameters. h is constructed by the average of word vector w_i extracted from WD and document vector d_j extracted from DC. We train all papers with Distributed Memory version of Paragraph Vector (PV-DM) Model [22] using a hidden layer neural network and get papers in the latent space $\mathbb{R}^k$, where similar papers will be close to each other.

3.2 Structure Learning

The second step is to acquire the structural vector for every paper $v_i \in G$ based on the Struc2vec algorithm [23] in the citation network. In order to get the structure representations of papers, it is necessary to measure the structural similarity between two papers without using other paper or edge attributes. Let $D_d(x)$ denotes the sets of papers from x to the target paper at the distance of d that means taking the papers from the paper x having less than d steps along the link path into account. $D_1(x)$ means the set of neighbours of x and $s(S)$ represents the ordered degree sequence of a set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

S of papers. The structural similarity measurement in [23] considering the hierarchy network is defined,

$$f_d(x,y) = f_{d-1}(x,y) + g(s(D_d(x)), s(D_d(y))), \\ k \geq 0 \; and \; |D_d(x)|, |D_d(y)| > 0 \quad (8)$$

where function $f_d(x,y)$ is the structural distance between x and y considering their d-hop neighbourhood. $g(s(D_d(x)), s(D_d(y))) \geq 0$ calculates the distance between the ordered sequences $s(D_d(x))$ and $s(D_d(y))$, and the initial f_{-1} is defined to 0. It should be noted that both x and y d-hop neighbours are accessible.

Dynamic Time Warping (DTW) is adopted to measure the structural distance between $s(D_d(x))$ and $s(D_d(y))$, two ordered degree sequences. DTW is a typical optimization problem which can handle different size sequences when matching with the stored model. The unknown sequence is unevenly warped or kinked to make the feature and template feature correct. Thus, DTW is a very powerful measure to improve the efficiency and accuracy of recognition the sequences.

Next, we need to calculate the similarity between papers in the multilayer graph MG, where d denotes each layer and is used to find d-hop neighbourhood of the papers. When x and y are in the same layer, the structural distance F in graph MG is

$$F_d(x,y) = \exp(f_d(x,y)). \quad (9)$$

Otherwise, the structural distance F of x in layer d connected to the corresponding x in layer $d-1$ and $d+1$ is

$$F(x_d, x_{d-1}) = 1, \\ F(x_d, x_{d+1}) = \log(\omega_d(x) + e), \quad (10)$$

where $\omega_d(x)$ is the number of edges incident to x that have larger F than the average F in layer d:

$$\omega_d(x) = \sum_{y \in V} (F_d(x,y) > \overline{F_d}). \quad (11)$$

$\omega_d(x)$ reflects the similarity of paper x to all other papers in layer d.

Like most network embedding methods, random walk is used to generate a sequence of papers. Considering different pair of papers (x,y) having different structural distance F in the multilayer graph MG, a biased random walk is suited for selecting the next paper from the current layer or different layer [23]. The probability that random walk decides to change layers is

$$Pr(x_d, x_{d+1}) = \frac{F(x_d, x_{d+1})}{F(x_d, x_{d+1}) + F(x_d, x_{d-1})}, \\ Pr(x_d, x_{d-1}) = 1 - Pr(x_d, x_{d+1}). \quad (12)$$

After gaining the paper sequences, we apply the Skip-Gram model to train these sequences, which is similar to Section 3.1.

3.3 Bridging Weighted-Citation Graph with Text and Structure

When we gain each paper's vector representation of text information and citation network structure, the following task is to integrate the text information and structure information into the citation network. Inspired by [21], we create the artificial text-based edges and structure-based edges in the citation network, which is shown in Figure 5.

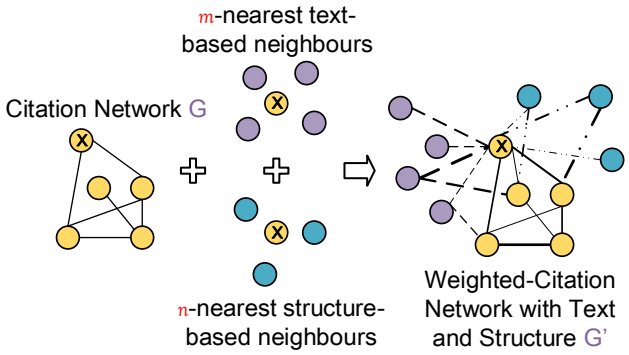


Fig. 5. Weighted-citation network with text and structure. Yellow nodes consist of the original citation network, purple nodes are m-nearest ($m = 4$) text-based neighbours of paper x and blue nodes are n-nearest ($n = 3$) structure-based neighbours of paper x. For building G', we connect x with purple nodes and blue nodes.

All the papers in G are represented as text-based vectors and structure-based vectors in $\mathbb{R}^k$ as mentioned in Section 3.1 and Section 3.2. We suppose these recommended papers based on the similarity, so we select m-nearest text-based neighbours (most similar) and n-nearest structure-based neighbours in $\mathbb{R}^k$ for each paper $v_i \in V$. Then, we add these text-based edges E_1 and structure-based edges E_2 to G and form a new text-based and structure-based citation network $G' = (V, E \bigcup E_1 \bigcup E_2, W)$.

We believe that adding similar text-based and structure-based edges can better discover the potential similarity between papers. However, how to assign different weights to each edge becomes a problem. In new network G', there are three different kinds of edges including similar text relationships, similar structure relationships, and citation relationships. First, we assign the initial weight 0.5 for each new edge. For example, paper v_j is the m-nearest text-based neighbours of paper v_i, then the weight of edge $e_{i,j}$ is 0.5. While paper v_j is the n-nearest structure-based neighbours of paper v_i at the same time, the weight of edge $e_{i,j}$ is added 0.5 to the original value, $0.5 + 0.5 = 1.0$. What mentioned above is the way to deal with the weight $w_{i,j}$ of similar text relationships and similar structure relationships. For citation relationships, we also assign the initial weight 0.5 for each new edge. What's more, we also consider the Amsler [24] similarity between two papers with citation relationship. The Amsler similarity between paper v_i and paper v_j is defined as:

$$amsler(v_i, v_j) = \frac{|(P_{v_i} \bigcup C_{v_i}) \bigcap (P_{v_j} \bigcup C_{v_j})|}{|(P_{v_i} \bigcup C_{v_i}) \bigcup (P_{v_j} \bigcup C_{v_j})|}, \quad (13)$$

where P_{v_i} represents the references of paper v_i and C_{v_i} represents the citations of paper v_i. We do not treat cited link and citing link differently, which means $w_{i,j} = w_{j,i}$. Thus, the more citation relationships v_i and v_j have in common, the more similar they are. We add the Amsler similarity to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

the initial weight 0.5 to assign a new weight $w_{i,j}$ for each citation relationship.

3.4 Learning from Graph

Now, we get the weighted-citation network with text and structure G'. For each v_i in G', we use the random walks on the weighted network G' according to the weight $w_{i,j}$. The probability of stepping from paper v_i to paper v_j is:

$$p(v_i, v_j) = w_{v_i,v_j} / \sum_{z \in C_i, z \neq i} (w_{v_i,v_z}), \tag{14}$$

where C_i is the set of papers that are the neighbours of paper v_i. The random walk is rooted at the paper v_i and chosen at the probability defined in Equation 14 from the neighbours of paper v_i. Then, a stochastic sequence with random variables $\{S^1_{v_i}, S^2_{v_i}, \cdots, S^l_{v_i}\}$ is generated. The method to convert a graph into a series of sequences is motivated by DeepWalk [25].

In order to gain the representation of papers in network G', we want to predict paper v_j, the neighbour of paper v_i, in the graph according to its feature representation. Similar to Section 3.1, we try to optimize the likelihood function that aims to maximize the average log-probability in Equation 15:

$$\sum_{v_i, v_j \in win}^{|win|} \log p(v_i|v_j), \tag{15}$$

where $p(v_i|v_j)$ can be rewritten as:

$$p(v_i|v_j) = \prod_{t=2}^{l^{v_i}} p(d^{v_i}_t | \boldsymbol{x}_{v_i}, \theta^{v_i}_{t-1}), \tag{16}$$

where

$$p(d^{v_i}_t | \boldsymbol{x}_{v_i}, \theta^{v_i}_{t-1}) = \begin{cases} \sigma(\boldsymbol{x}^{\mathrm{T}}_{v_i} \theta^{v_i}_{t-1}) & d^{v_i}_t = 0; \\ 1 - \sigma(\boldsymbol{x}^{\mathrm{T}}_{v_i} \theta^{v_i}_{t-1}) & d^{v_i}_t = 1, \end{cases} \tag{17}$$

and l represents the number of papers in the path $\{S^1_{v_i}, S^2_{v_i}, \cdots, S^l_{v_i}\}$ in the binary Huffman tree, $d^{v_i}_t$ represents the Huffman code (the value is 0 or 1) of paper v_i and $\theta^{v_i}_{t-1}$ represents the t-th paper vector in the path $\{S^1_{v_i}, S^2_{v_i}, \cdots, S^l_{v_i}\}$. Both sets of negative and positive (v_i, v_j) are taken to approximate Equation 15 based on the idea of hierarchical softmax. Then, Equation 15 can be rewritten as:

$$L(v_j, v_i, t) = (1 - d^{v_i}_t) \cdot \log[\sigma(\boldsymbol{x}^{\mathrm{T}}_{v_i} \theta^{v_i}_{t-1})] + d^{v_i}_t \cdot \log[1 - \sigma(\boldsymbol{x}^{\mathrm{T}}_{v_i} \theta^{v_i}_{t-1})]. \tag{18}$$

The object function of Equation 18 can be optimized by stochastic gradient descent algorithm and the $\boldsymbol{x}_{v_i}$ is what we need the representations for paper v_i.

For the paper recommendation task, we want to find the most similar papers for each target paper. Since we get the vector representation $\boldsymbol{x}_{v_i}$ of every paper with the latent space $\mathbb{R}^k$, we can calculate the cosine similarity between each pair papers with Equation 19:

$$sim(\boldsymbol{x}_{v_i}, \boldsymbol{x}_{v_j}) = \frac{\boldsymbol{x}_{v_i} \cdot \boldsymbol{x}_{v_j}}{\sqrt{|\boldsymbol{x}_{v_i}| \cdot |\boldsymbol{x}_{v_j}|}}. \tag{19}$$

TABLE 2
Data set statistics

Statistics information	Data
Number of citing papers	393
Number of reference papers	1760
Number of citation relationships	2664
Average number of reference papers of each citing paper	6.78 (max. 20)

For each target paper, we rank the set of candidate papers in order of decreasing similarity and find top-Q similar papers to recommend.

4 Experiments

In this section, we describe the performance of our proposed framework for research paper recommendation. We introduce the data set and experimental setup for our study, as well as the evaluation metrics and baseline methods used to compare our framework. We also present and analyze the results of our experiments.

4.1 Data set

We utilize PRA (Physical Review A) from APS[1] (American Physical Society) that is well known Physics journals libraries in our experiments. We use papers of PRA from 2007 to 2009 including 9151 citation relationships between 7547 papers. We also remove papers which cited less than 5 papers. We build a citation network based on the rest of papers and relationships, and find the maximal connected component subgraphs of this citation network as our experimental citation network G. Finally, we screen out 2153 papers from PRA including 393 papers that have reference papers and 2664 citation relationships. The distribution of the preprocessed data set is shown in Table 2.

4.2 Experimental Setup

To evaluate the quality of recommendations, we divide data set randomly into a training set and a test set using the following criterion. For each citing paper, we randomly select it's reference papers into the test set with the ratio of O. The training set is treated as known information used by our framework for learning representation $\boldsymbol{x}_{v_i}$ of each paper, while the test set is regarded as unknown information used for testing the performance of recommendation results. The text information that is needed in Section 3.1 includes paper titles of every paper in training set. We use CountVectorizer from scikit-learn [26] to extract 2630 words and build a binary vector of 2630 dimensions (i.e., '1' in the vector represents that the word appears in this paper title) for each paper. We set the same vector dimension k for textual vector and structural vector as final paper vector we need. We select those papers which cited more than 7 papers in citation network G as Target Papers. For every target paper, we create a recommendation list of length Q to find the top-Q similar papers.

1. https://journals.aps.org/datasets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

4.3 Baseline Methods

4.3.1 Doc2vec

Doc2vec [22] is a text-only method to represent the text as a vector using a three-layer neural network as the framework. PV-DM Model we used in Section 3.1 uses contexts around the target word as input and outputs the target word after weighting at the mapping layer.

4.3.2 Struc2vec

Struc2vec [23] is representation learning method that captures structural identity of nodes in the network considering the structural similarity and multilayer graph.

4.3.3 DeepWalk

Deepwalk [25] is a network-only representation learning method. Deepwalk takes random walk paths from network as sentences and nodes as words to learn the node representations by applying the Skip-Gram algorithm.

4.3.4 CCF

CCF [8] (Context-Based Collaborative Filtering) is a context-based collaborative filtering citation recommendation method that obtains the paper representation of each citing paper from the citation context by applying the Collaborative Filtering algorithm.

For first three embedding methods, we use the same parameters listed below.

- The walk length wl: 40;
- The window size $|win|$: 10;
- The size of negative samples: 5.

4.4 Evaluation Metrics

To evaluate the performance of our proposed framework VOPRec, we employ three popular metrics that are widely used in the fields of recommendation systems and one ranked information retrieval (IR) evaluation metric that can measure the importance of top ranked recommended papers: Precision, Recall, F1, and normalized discounted cumulative gain (NDCG) [27].

4.4.1 Precision

Precision is the probability that the papers in the recommendation list are the reference papers of target papers in our test set. For each target paper v_i,

$$precision_{v_i} = \frac{Q^i_{rp}}{Q}, \tag{20}$$

where Q^i_{rp} denotes the number of papers in the recommendation list that are the reference papers of target paper v_i and Q is the length of recommendation list. The whole paper recommendation's precision is

$$precision = \frac{1}{\alpha}\sum_{i=1}^{\alpha} precision_{v_i}, \tag{21}$$

where α is the number of target papers. The higher precision, the higher accuracy of the recommendation method.

4.4.2 Recall

Recall is the probability that recommended papers for target papers appear in target papers' references papers list. For each target paper v_i,

$$recall_{v_i} = \frac{Q^i_{rp}}{Q^i_p}, \tag{22}$$

where Q^i_p is the number of references papers of target paper v_i in the test set. The whole paper recommendation's recall, similar to precision, is

$$recall = \frac{1}{\alpha}\sum_{i=1}^{\alpha} recall_{v_i}. \tag{23}$$

4.4.3 F1

With the increasing of the length recommendation list Q, it is hard to get higher precision and higher recall at the same time. A larger value of Q generally gives a higher recall but a lower precision, while F1 is weighted mean of precision and recall. The whole paper recommendation's F1 is

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{24}$$

4.4.4 NDCG

Discount cumulative gain (DCG) is well suited to evaluate the recommendation system, as it gives a higher score to the relevant items that appear on the top of the ranked results than those ranked lower. For each target paper v_i, DCG_{v_i} is calculated as

$$DCG_{v_i} = \sum_{Q}^{j=1} \frac{2^{r(j)} - 1}{\log(1+j)}, \tag{25}$$

where $r(j)$ is a relevance level of results returned at the rank $j(j = 1, 2, \cdots, Q)$. In our case, $r(j) = 1$ means the recommended paper is the reference paper of target paper in the test set, while $r(j) = 0$ is not the true reference paper of target paper. NDCG is the normalization of DCG. For each target paper v_i,

$$NDCG_{v_i} = \frac{DCG_{v_i}}{IDCG_{v_i}}, \tag{26}$$

where $IDCG_{v_i}$ is the ideal ranking in which the most relevant items are ranked on the top (i.e., given a ranking $(0, 0, 1, 1, 0)$, the ideal ranking is $(1, 1, 0, 0, 0)$). The whole paper recommendation's NDCG is

$$NDCG = \frac{1}{\alpha}\sum_{i=1}^{\alpha} NDCG_{v_i}. \tag{27}$$

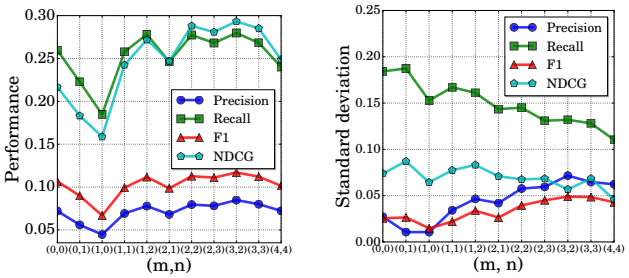
4.5 Results Analysis

4.5.1 Parameter Sensitivity

In skip-gram-based representation learning models, there are several common parameters. We conduct a sensitivity analysis of VOPRec to three important parameters: the number of nearest text-based neighbours m, the number of nearest structure-based neighbours n, and vector dimension k. We vary each of them and fix the others for examining the parameter sensitivity of VOPRec.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

First, we set the vector dimension k fixed as 100 and select eleven pairs parameters of m and n. Figure 6 shows the comparison results of VOPRec, where (m, n) are equal to $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$, $(1, 2)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(3, 2)$, $(3, 3)$, and $(4, 4)$, respectively. We calculate the average precision, recall, F1, and NDCG of recommendation list that ranges from 1 to 20. From Figure 6(a), we can see that the results of $(m, n) = (3, 2)$ are 90.30%, 51.42%, 75.14%, 84.69% higher than the worst result in precision, recall, F1, and NDCG, respectively. We can find out that only text information or topology structure (i.e., $(m, n) = (0, 1)$ or $(m, n) = (1, 0)$) has low precision, recall, F1, and NDCG, which is even lower than $(m, n) = (0, 0)$. Especially, when (m, n) is $(0, 0)$, we only consider the weighted citation network, which is a little different from the baseline method DeepWalk used on the un-weighted citation network. Figure 6(b) shows the standard deviation for different length of recommendation list. With the increase of m and n, recall and NDCG are more stable while precision and F1 are not. Since our experimental data set is not large enough, too many text-based or structure-based neighbours added to the citation network will reduce the impact network topology so as to reduce the performance of VOPRec on the contrary. To achieve a better performance, we set $(m, n) = (3, 2)$ in the following results analysis.



(a) (b)

Fig. 6. Evaluation results on different (m, n). We set dimension k as 100 and the ratio of test set O is 20%. (a) shows the average performance, and (b) shows the standard deviation.

Considering the small probability for two papers citing each other with far distance in the network but similar network structure, we analyze the length of shortest path between each paper and its n-nearest structure-based neighbours. We regard the $link = 1$ as papers that are the reference papers of target papers in the test set. We exclude the inexistent path between two papers when we calculate the average length of shortest path for $link = 0$ and all n neighbours. From table 3, we can find that the average length of shortest path for $link = 0$ is much shorter than $link = 1$. Thus, there is a small probability for two papers citing each other if they are too far away in the network. To reduce the impact of these too far neighbours, we select each paper's n-nearest structure-based neighbours whose shortest path to this paper is shorter than the average length of shortest path in table 3 for different ratio of test set O. We calculate the average precision, recall, F1, and NDCG of recommendation list that ranges from 1 to 5 with fixed vector dimension $k = 150$ and $(m, n) = (3, 2)$. Table 4 shows the performance improvement of selected neighbours than just similar neighbours. We can find out that the performance of VOPRec improves a lot when the ratio of test set O is over 50%. The criteria for selecting neighbours according to the shortest path can be adjusted for some situations.

TABLE 3
Shortest path for different test set and n-nearest structural neighbours

O	n	Average shortest path for $link = 0$	Average shortest path for $link = 1$	Average shortest path	Proportion of inexistent path (%)
20%	1	5.5	2.0	5.2	5%
	2	5.5	2.0	5.3	6%
	3	5.8	2.0	5.6	7%
	4	6.0	2.0	5.8	7%
30%	1	6.3	2.0	5.9	17%
	2	6.4	2.0	6.0	16%
	3	7.2	2.0	6.9	16%
	4	7.7	2.0	7.3	16%
40%	1	6.2	2.0	5.9	19%
	2	6.3	2.0	5.9	21%
	3	6.5	2.0	6.2	23%
	4	6.7	2.0	6.4	25%
50%	1	4.2	2.0	3.8	24%
	2	4.6	2.1	4.4	27%
	3	4.6	2.4	4.4	30%
	4	5.0	2.4	4.7	35%
60%	1	3.7	2.0	3.4	57%
	2	3.8	2.1	3.4	58%
	3	4.0	2.1	3.7	62%
	4	4.2	2.3	3.9	65%
70%	1	3.5	2.0	3.4	57%
	2	3.8	2.0	3.6	63%
	3	4.0	2.5	3.8	68%
	4	4.1	2.4	3.8	70%

TABLE 4
Performance improvement for selected n-nearest structural neighbours

Performance improvement (%)		Metrics			
		Precision	Recall	F1	NDCG
O	0.2	81%	104%	92%	107%
	0.3	52%	59%	57%	57%
	0.4	64%	61%	62%	59%
	0.5	39%	46%	44%	38%
	0.6	126%	112%	114%	120%
	0.7	115%	127%	123%	122%

Then, we let the vector dimension k range from 50 to 300 and calculate the average precision, recall, F1, and NDCG of recommendation list that ranges from 1 to 20 for each k. It can be seen from Figure 7 that the result of VOPRec is the best when k is 150 in general. However, there is not a significant difference when k ranges from 100 to 200. Since our experimental data set is not large enough, k is kept constant at 150. Bigger networks containing more data may need larger k to capture all of data.

4.5.2 *Effects of Recommendation List*

We want to evaluate the performance of VOPRec with the increase of recommendation list length Q and the fixed dimension $k = 150$. Figure 8 shows comparsions between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

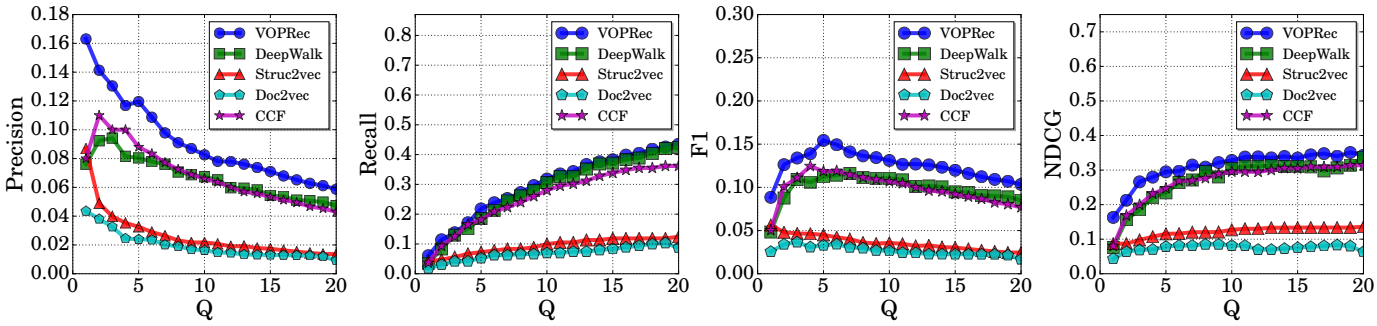


Fig. 8. Comparison between VOPRec and other methods in recommendation quality over different length of recommendation list Q. We set k as 150 for all embedding methods, ratio of test set O as 20% and (m, n) as $(3, 2)$.

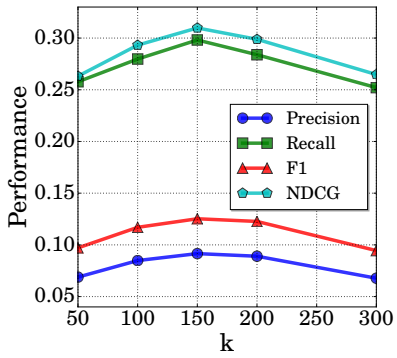


Fig. 7. Evaluation results on different dimension k. We set (m, n) as $(3, 2)$ and the ratio of test set O is 20%.

TABLE 5
Performance improvement for VOPRec

Performance improvement (%)		Metrics			
		Precision	Recall	F1	NDCG
Methods	DeepWalk	19%-114%	1%-72%	16%-83%	2%-114%
	CCF	17%-104%	4%-66%	12%-76%	9%-104%
	Struc2vec	88%-346%	46%-258%	57%-335%	88%-172%
	Doc2vec	271%-535%	231%-400%	246%-519%	275%-441%

VOPRec and baseline methods in terms of precision, recall, F1, and NDCG.

With the increase of Q, the precision of all methods decreases and the recall of all methods increases, which is the general phenomenon for most recommendation systems. F1 of most methods (except Struc2vec) tends to increase at first then decrease with the increase of Q. NDCG of all methods has a little increase when $Q \leq 5$ and remains stable when $Q \geq 5$. VOPRec achieves the best performance compared with baseline methods of all Q. Table 5 shows the performance improvement of VOPRec compared with DeepWalk, CCF, Struc2vec, and Doc2vec. DeepWalk and CCF have the similar performance for they are based on the citation relationship between two papers. Doc2vec has the worst performance indicates that the citation relationship between two papers is not only related to the text similarity.

4.5.3 Effects of Test Set

For each target paper, we randomly select 20%, 30%, 40%, 50%, 60%, and 70% reference papers as the test set, the fixed recommendation list $Q = 3$ and fixed dimension $k = 150$ to evaluate the performance of VOPRec. Figure 9 shows comparsions between VOPRec and baseline methods in terms of precision, recall, F1 and NDCG.

With the increasing fractions of test set, the precision and NDCG of VOPRec fluctuates a little while recall and F1 take on a descending trend. The precision, recall, F1, NDCG of CCF decrease with the increase of ratio of test set. Surprisingly, the precision, F1, NDCG of DeepWalk have a little decrease at first and increase with the increase of ratio of test set O. While the changes of fractions of test set have a little influence on Doc2vec and Struc2vec because these two methods have less dependence on the whole network topology. VOPRec outperforms other methods over all ratios of test set in general. Network-based method, DeepWalk has the second best performance. Although Doc2vec and Struc2vec are not sensitive to the ratio of test set, the performance of these two methods are not so good. In conclusion, VOPRec needs more data to train in order to get a better performance.

5 Related Work

In the age of big data, scholarly data set is becoming bigger and bigger, which may result in the information overload [1]. Academic recommendation systems have been developed to solve such problem by providing scholars with relevant and suitable items, e.g., collaborators [28], venues [29], and papers [8]. In the field of paper recommendation, previous work can be mainly divided into three categories based on their methods, including content-based filtering, collaborative filtering, and graph-based recommendations [3].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2018.2830698, IEEE Transactions on Emerging Topics in Computing

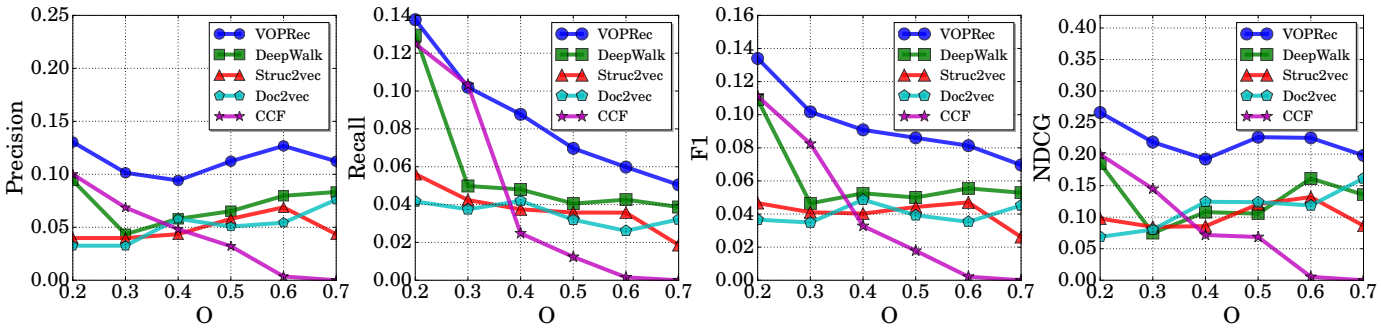


Fig. 9. Comparison between VOPRec and other methods in recommendation quality over different ratios of test set O. We set k as 150 for all embedding methods and length of recommendation list Q as 3.

5.1 Content-based Paper Recommendation

In the area of paper recommendation, content-based filtering is the most common and basic recommendation technique. The central component of content-based methods is the scholar profiling process, where the interests of scholars are inferred from their publication content. To generate a recommendation list, the similarity between scholar profile and recommendation candidates is calculated by vector space model or cosine similarity coefficient so that top-k items can be gained for recommendation.

Content-based approaches profile scholars' interest by extracting words from the paper title, abstract, keywords, or from the papers' main body based on the information provided by the data set. Most of these approaches use the plain words as features. Moreover, some extend plain words to n-gram [30], topics [5], and concepts [31] which are extracted by LDA (Latent Dirichlet Allocation). One of the typical content-based paper recommendation is proposed by Sugiyama and Kan [4]. Their main hypothesis is that an author's previous published works show the latent interests of a scholar. The core part of their model is to enhance the scholar profile derived directly from previous publication by considering information coming from both the past papers' referenced papers as well as these papers' citations. Meanwhile, they differentiate between junior researchers who only publish one paper and senior researchers who have more than one papers.

Content-based paper recommendation method allows a scholar-based modelling so that the recommendation can achieve a high accuracy for each scholar individually. Meanwhile, it can generate scholar model automatically so that less up-font classification work is needed. However, the method is limited in low serendipity and overspecialization whereas it leads to recommend papers that scholars already know. The quality and popularity of recommended papers are also overlooked in most content-based paper recommendation methods [3].

5.2 Collaborative Filtering-based Paper Recommendation

Collaborative filtering is one of the most successful recommendation technique to data [32], [33]. The basic assumption of collaborative filtering is to recommend items based on the opinions of other like-minded users, where two users are considered like-minded when they rate similar or same items. Once the like-minded users are identified, items positively rated by these users will be recommended to the target users [8], [34], [35]. The biggest advantage of collaborative filtering-based recommendation compared with content-based method is that it is content independent, i.e., no item information is needed.

However, when applying collaborative filtering into paper recommendation, the biggest challenge is the "cold-start" problem because scholars seldom rate papers they read or cite. However, paper ratings play a significant role in collaborative filtering-based recommendation methods. Although some reference management tools such as Endnote [36] provides such rating service, such data set is too small to generate accuracy recommendation results. In order to solve the cold-start problem in utilizing collaborative filtering techniques, some works propose to take advantages of implicit ratings inferred from the interactions between scholars and papers. For example, McNee et al. [37] propose that an scholar's citation denotes a positive vote for a paper. Thus, if two scholars cite the same paper, they are like-minded. Similarly, if a scholar reads or cites a paper, the citations of the cited papers may be interesting for the scholar. Liu et al. [8] propose a context-based collaborative filtering method for paper recommendation by assuming that if two citing papers are significantly co-occurring with the same cited paper(s), they should be similar to some extent.

Compared with content-based recommendation methods, collaborative filtering-based paper recommendation methods can generate more serendipitous results since they calculate scholar similarity instead of paper similarity. However, collaborative filtering-based methods need more computing time and more offline data processing.

5.3 Graph-based Paper Recommendation

Graph-based paper recommendations usually construct a citation network where two papers are connected if one paper cites another. Based on the constructed citation network, the paper recommendation task can be regarded as citation link prediction problem [38], [39]. Furthermore, some works consider heterogenous academic networks by considering

various entities including authors, venues, and papers [40], [41].

The key of citation link prediction is to calculate the similarity between two given papers. Strohman et al. [42] design a paper recommendation system where the relevance between two papers is measured by a combination of text features and citation graph features. They find that the most important features are similarity between bibliographies of papers and Kate distance in network. To alleviate the network sparsity problem, Sugiyama et al. [6] identify "potential citation papers" in citation networks by using the collaborative filtering method. Meanwhile, they investigate which section of the article can be leveraged to represent papers since different sections play different roles in a paper.

In graph-based paper recommendations, previous works mainly extract structural information such as degree or common neighbours to calculate paper similarity. However, such hand-engineered features are inflexible and designing such features may be time-consuming process. Recently, network representation methods have been proposed to learn representations that encode structural information about the graph for citation recommendation [14], [21].

6 Conclusion

In the era of scholarly big data, it is always difficult for scholars to find relevant papers from millions of publications. In this paper, we aim to recommend papers based on network representation learning technique when scholars read or cite the target papers. To this end, we propose VOPRec, which considers both paper content and network structure. Specifically, VOPRec learns textual vectors of paper from paper content, e.g., title and abstract to find potential papers of similar research interest. Then, the structural vectors of papers are learned from structural identity in order to find potential papers of similar structure. VOPRec proposes to bridge these two vectors with weighted citation network by connecting m-nearest text-based neighbours and n-nearest structure-based neighbours. Based on the reconstructed weighted citation network, the paper similarity can be calculated with network embeddings for scientific paper recommendation. We further show the effectiveness of VOPRec on a real-world data set by comparison with state-of-the-art paper recommendation methods and results show that VOPRec outperforms these baseline methods in terms of precision, recall, F1, and NDCG.

In the future, we will extend VOPRec to heterogeneous networks which contains not only papers but also others academic entities such as authors, conferences, and journals. Meanwhile, we will test the effectiveness of VOPRec on other data sets to show its universal applicability.

References

[1] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big scholarly data: A survey," *IEEE Transactions on Big Data*, vol. 3, no. 1, pp. 18–35, 2017.

[2] S. Khan, X. Liu, K. A. Shakil, and M. Alam, "A survey on scholarly data: From big data perspective," *Information Processing & Management*, vol. 53, no. 4, pp. 923–944, 2017.

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

[4] K. Sugiyama and M.-Y. Kan, "Scholarly paper recommendation via user's recent research interests," in *Proceedings of the 10th annual joint conference on Digital libraries*. ACM, 2010, pp. 29–38.

[5] Y. Jiang, A. Jia, Y. Feng, and D. Zhao, "Recommending academic papers via users' reading purposes," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 241–244.

[6] K. Sugiyama and M.-Y. Kan, "Exploiting potential citation papers in scholarly paper recommendation," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2013, pp. 153–162.

[7] F. Xia, H. Liu, I. Lee, and L. Cao, "Scientific article recommendation: Exploiting common author relations and historical preferences," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 101–112, 2016.

[8] 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.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[10] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques and applications," *arXiv preprint arXiv:1709.07604*, 2017.

[11] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *arXiv preprint arXiv:1705.02801*, 2017.

[12] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 135–144.

[13] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information." in *IJCAI*, 2015, pp. 2111–2117.

[14] H. Tian and H. H. Zhuo, "Paper2vec: Citation-context based document distributed representation for scholar recommendation," *arXiv preprint arXiv:1703.06587*, 2017.

[15] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[16] F. Menczer, "Combining link and content analysis to estimate semantic similarity," in *International World Wide Web Conference on Alternate Track Papers & Posters*, 2004, pp. 452–453.

[17] S. Gao, L. Denoyer, and P. Gallinari, "Temporal link prediction by integrating content and structure information," in *ACM International Conference on Information and Knowledge Management*, 2011, pp. 1169–1174.

[18] D. R. Amancio, O. N. O. Jr, and L. D. F. Costa, "Three-feature model to reproduce the topology of citation networks and the effects from authors visibility on their h-index," *Journal of Informetrics*, vol. 6, no. 3, pp. 427–434, 2012.

[19] F. N. Silva, D. R. Amancio, M. Bardosova, L. D. F. Costa, and O. N. O. Jr, "Using network science and text analytics to produce surveys in a scientific topic," *Journal of Informetrics*, vol. 10, no. 2, pp. 487–502, 2016.

[20] H. Trevor, T. Robert, and F. Jerome, "The elements of statistical learning," *Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2009.

[21] S. Ganguly and V. Pudi, "Paper2vec: Combining graph and text information for scientific paper representation," in *European Conference on Information Retrieval*, 2017, pp. 383–395.

[22] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," vol. 4, pp. II–1188, 2014.

[23] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *The ACM SIGKDD International Conference*, 2017, pp. 385–394.

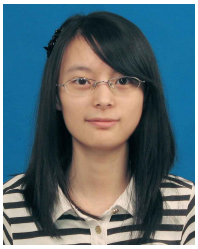
[24] R. A. Amsler, "Application of citation-based automatic classification," 1972.

[25] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot,

and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[27] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, pp. 41–48.
[28] F. Xia, Z. Chen, W. Wang, J. Li, and L. T. Yang, "Mvcwalker: Random walk-based most valuable collaborators recommendation exploiting academic factors," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 364–375, 2014.
[29] F. Xia, N. Y. Asabere, H. Liu, Z. Chen, and W. Wang, "Socially aware conference participant recommendation with personality traits," *IEEE Systems Journal*, 2014.
[30] F. Ferrara, N. Pudota, and C. Tasso, "A keyphrase-based paper recommender system." in *IRCDL*. Springer, 2011, pp. 14–25.
[31] S. Bethard and D. Jurafsky, "Who should i cite: learning literature search models from citation behavior," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 609–618.
[32] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.
[33] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender systems handbook*. Springer, 2015, pp. 77–118.
[34] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 473–480.
[35] A. Vellino, "A comparison between usage-based and citation-based methods for recommending scholarly research articles," *Proceedings of the Association for Information Science and Technology*, vol. 47, no. 1, pp. 1–2, 2010.
[36] J. Evans and B. Davies, "14 endnote: the embodiment of consciousness," *Body knowledge and control: Studies in the sociology of physical education and health*, p. 207, 2004.
[37] 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 *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. ACM, 2002, pp. 116–125.
[38] O. Küçüktunç, E. Saule, K. Kaya, and Ü. V. Çatalyürek, "Recommendation on academic networks using direction aware citation analysis," *arXiv preprint arXiv:1205.1143*, 2012.
[39] Y. Liang, Q. Li, and T. Qian, "Finding relevant papers based on citation relations," *Web-age information management*, pp. 403–414, 2011.
[40] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles, "Learning multiple graphs for document recommendations," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 141–150.
[41] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine learning*, vol. 81, no. 1, pp. 53–67, 2010.
[42] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach, "Recommending citations: translating papers into references," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1910–1914.

Xiangjie Kong (M'13-SM'17) received the BSc and PhD degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor in School of Software, Dalian University of Technology, China. He has served as (Guest) Editor of several international journals, Workshop Chair or PC Member of a number of conferences. Dr. Kong has published over 50 scientific papers in international journals and conferences (with 30+ indexed by ISI SCIE). His research interests include social computing, mobile computing, and computational social science. He is a Senior Member of IEEE and CCF, and a Member of ACM.



Mengyi Mao received her B.S. degree in network engineering from Yunnan University, Kunming, China, in 2016. She is currently pursuing the M.Sc. degree in Software Engineering in Dalian University of Technology, Dalian, China. Her research interests include big scholarly data, collaboration network analysis and computational social science.

Wei Wang received his B.S. degree in Electronic Information Science and Technology from Shenyang University, Shenyang, China, in 2012. He is currently working toward the Ph.D. degree in Software Engineering in Dalian University of Technology, Dalian, China. His research interests include big scholarly data, social network analysis and computational social science.

Jiaying Liu received the BS degree in software engineering from Dalian University of Technology, China, in 2016. She is currently working toward the masters degree in the School of Software, Dalian University of Technology, China. Her research interests include big scholarly data, social network analysis, and science of success.

Bo Xu received the BSc and PhD degrees from the Dalian University of Technology, China, in 2007 and 2014, respectively. She is currently a lecture in School of Software at the Dalian University of Technology. Her current research interests include social computing, machine learning, literature data mining, and natural language processing.