

# Academic Paper Recommendation Based on Community Detection in Citation-Collaboration Networks

Qisen Wang<sup>1</sup>, Wenzhong Li<sup>1,2,3</sup>(✉), Xiao Zhang<sup>1</sup>, and Sanglu Lu<sup>1,2,3</sup>

<sup>1</sup> State Key Laboratory for Novel Software Technology,  
Nanjing University, Nanjing, China  
lwz@nju.edu.cn

<sup>2</sup> Sino-German Institutes of Social Computing, Nanjing University, Nanjing, China

<sup>3</sup> Collaborative Innovation Center of Novel Software Technology  
and Industrialization, Nanjing, China

**Abstract.** Academic search engine plays an important role for science research activities. One of the most important issues of academic search is paper recommendation, which intends to recommend the most valuable literature in a domain area to the users. In this paper, we show that exploring the relationship of collaboration between authors and the citation between publications can reveal implicit relevance between papers. By studying the community structure of the citation-collaboration network, we propose two paper recommendation algorithms called Adaptive and Random Walk, which comprehensively consider several metrics such as textural similarity, author similarity, closeness, and influence for paper recommendation. We implement an academic paper recommendation system based on the dataset from Microsoft Academic Graph. Performance evaluation based on the assessments of 20 volunteers show that the proposed paper recommendation methods outperform the conventional search engine algorithm such as PageRank. The efficiency of the proposed algorithms are verified by evaluation.

**Keywords:** Paper recommendation · Social network · Citation-collaboration network · Community detection

## 1 Introduction

Searching for academic literature is of great important for researchers and engineers. As for the development of search engine theories and practices, scholar search engine such as Google Scholar [1], Microsoft Academic Search [2], and ArnetMiner [3], which specify the data source to science domain, are also built as more personalized means to search according to the content and influence of a publication. Most of the existing academic search engines rank the importance of a paper based on the relevance of keywords, the citation counts and influence index. However, such keyword-based search did not consider the collaboration

between authors and the citation between papers, and sometimes cannot satisfy the need of academic paper recommendation, where the user hopes the search engine to recommend the most valuable literature in a domain area. For example, an author works on “natural language processing” may has coauthors who works on “information retrieval”, which suggests that the areas of “natural language processing” and “information retrieval” maybe highly relevant. But a keyword based search without considering author collaborations cannot reveal such relevance. Another example is that the papers on “information diffusion” may frequently cite the papers on “epidemic model”, which suggest that “epidemic model” could be quite valuable for “information diffusion”, but a search engine without considering the citations between papers also fail to address such implicit correlations.

Therefore, to tackle the issues of academic paper recommendation, it is worthwhile to explore the relationship of collaboration and citation in literature. Given the large number of publications, such relationship can be described by a two-layer complex network called *citation-collaboration network*, where one layer is from the perspective of citation relations between publications, and the other is from the perspective of collaboration relations between authors. According to the homophily theory [4] of social networks, people are more likely to share their ideas with people from the same area, the same university, or at least speak the same language. Similar discipline also applies for the citation-collaboration network and forms the phenomenon so called “academic circles”. Community detection algorithms [5,6] can be used to identify the “academic circles”, which publications within the same community should be more relevant to each other, and publications from different communities are less relevant.

In this paper, we propose an academic paper recommendation system based on community detection in citation-collaboration networks. We first introduce a graph transform method to transform the complex network to a uniform directed graph with nodes representing papers and weights representing the closeness between them. Then we use a *weighted label propagation algorithm (WLPA)* to divide the citation-collaboration networks into smaller communities. After community detection, we can identify a group of relevant publications representing as a community and then perform paper recommendation algorithm in the same community. We propose two ranking algorithms called *Adaptive* and *Random Walk*, which comprehensively consider several metrics such as textural similarity, author similarity, closeness, and influence for paper recommendation. We implement an academic search engine based on the dataset from Microsoft Academic Graph [7]. Performance evaluation based on the volunteers’ assessments shows that the proposed paper recommendation methods outperform the conventional search engine algorithm such as PageRank. The efficiency of the proposed algorithms are verified by the evaluation.

## 2 Related Work

### 2.1 Community Detection

Community detection is a classical problem aiming at splitting a graph into a given number of groups while minimizing the cost of edge cut [5,8]. In early work, the community detection algorithms require some other parameters except graph data, such as community sizes or numbers. Later, the GN algorithm [6] proposed by Newman and Girvan made the community detection more independent with a divisive way to split the whole graph into groups by the edges with biggest betweenness. Some other alternative methods emerged in the past decades, where the similarity between vertices were used as a vital measurement for community detection. An example is the hierarchical clustering proposed by Newman [9]. Rosvall and Bergstrom proposed a community detection method based on information theory [10]. Fu et al. [11] proposed a memetic algorithm for community detection in networks based on the genetic algorithm.

To reduce the time complexity and achieve efficient community detection in massive social networks, Raghavan, Albert, and Kumar proposed the label propagation algorithm (LPA), which has a relatively low time complexity of  $O(n)$  [12]. However, LPA algorithm cannot be applied in directed weighted graphs, which is the focus of this paper.

### 2.2 Search Algorithms

Searching for web pages or academic papers in a network has been widely studied in the past. The most frequently adopted algorithm for search engine is the PageRank [13] algorithm from Google. The key idea is to initialize each node with a different value, and each nodes's value is given to its out links. After several rounds of iteration, each node's value will remain relatively stable. Therefore the importance of the nodes can be ranked according the value. Another well known search algorithm was the HITS algorithm proposed by Jon Kleinberg [14]. Taher Haveli-wala [15] proposed the Hilltop algorithm by putting the ideas of PageRank algorithm and HITS algorithm together. To the best of our knowledge, recommending academic paper based on community detection in citation-collaboration networks has not been studied in the past.

## 3 System Model and Problem Description

### 3.1 Citation-Collaboration Network

We introduce the citation-collaboration network to describe the citation among academic papers and the collaboration among authors. The citation-collaboration network is represented by a directed graph. A node in the graph represents a paper or an author. There are three different kinds of links in the graph representing different relationships: (1) citation relationship: if paper  $p_i$  is

cited by paper  $p_j$ , there is a directed link from  $p_j$  to  $p_i$ ; (2) collaboration relationship: if author  $a_i$  has a joint paper with author  $a_j$ , then there are bidirectional links between  $a_i$  and  $a_j$ ; (3) author-paper relationship: if paper  $p_i$  has an author  $a_j$ , then there is a directed link from  $a_j$  to  $p_i$ . The citation-collaboration network is also a weighted graph, where the citation relationship and the author-paper relationship have weight 1, while the collaboration relationship has a weight equals to the number of joint papers of two authors.

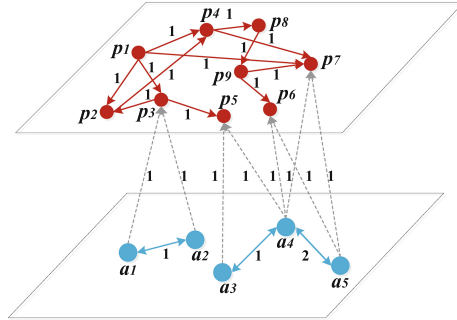


Fig. 1. A citation-collaboration network.

Figure 1 shows an example of a citation-collaboration network. It can be viewed as a two-layer complex networks: the citation network formed by citation between publications, and the collaboration network formed by joint works of authors or researchers. The cross-layer links indicate the author-paper mappings between the two layers.

### 3.2 Problem Description

Let  $G = \langle V_1, V_2, E_1, W_1, E_2, W_2, E_3, W_3 | E_1 \subseteq V_1 \times V_1, E_2 \subseteq V_2 \times V_2, E_3 \subseteq V_2 \times V_1 \rangle$  be a citation-collaboration graph, and  $p_i, p_j \in V_1$  are two papers. We use the following metrics to measure the relevance of the two papers.

**Definition 1 (Textural Similarity).** *The textural similarity of two papers is calculated by the Jaccard Coefficient of their keywords, which is defined as*

$$TextSim(p_i, p_j) = \frac{|Key(p_i) \cap Key(p_j)|}{|Key(p_i) \cup Key(p_j)|},$$

where  $Key(p_i)$  and  $Key(p_j)$  are the set of keywords abstracted from the two papers.

Similarly, we define the author similarity of two papers as the Jaccard Coefficient of their authors.

**Definition 2 (Author Similarity).**

$$AuthorSim(p_i, p_j) = \frac{|Author(p_i) \cap Author(p_j)|}{|Author(p_i) \cup Author(p_j)|},$$

where  $Author(p_i)$  and  $Author(p_j)$  are the set of authors of the two papers.

**Definition 3 (Closeness).** The closeness of two papers is defined as their normalized distance in the citation-collaboration graph. Specifically,

$$Closeness(p_i, p_j) = \frac{|Path(p_i, p_j)|}{\max_{\forall p_x, p_y \in V_1} |Path(p_x, p_y)|},$$

where  $Path(p_i, p_j)$  is the shortest path between  $p_i$  and  $p_j$  in the citation-collaboration graph,  $|Path(p_i, p_j)|$  indicates the number of hops in the path, and denominator indicates the maximum path length in the graph.

The influence of a paper is measured by its structural importance in the citation-collaboration graph. Intuitively, we can use the *Degree Centrality* [16] to define the influence of a paper, which equals to the total number of citations from other papers. The influence is indicated by the normalized degree centrality defined as follows.

**Definition 4 (Influence).** The influence of a paper  $p \in V_1$  is calculated by

$$Influence(p) = \frac{\sum_{\forall q \in V_1} w_{qp}}{\max_{\forall p' \in V_1} \sum_{\forall q \in V_1} w_{qp'}},$$

where  $w_{qp}$  is the indicator that  $w_{qp} = 1$  if there exists a link from node  $q$  to node  $p$  in  $G$  and  $w_{qp} = 0$  otherwise, and the denominator indicates the maximum citation count in the graph.

Given a paper  $p \in V_1$ , our objective is to find the most relevant and high influence papers. We rank the papers using a comprehensive function that considering the above metrics. For a paper  $q \in V_1, q \neq p$ , its ranking can be represented by

$$Rank(q, p) = f(TextSim(q, p), Closeness(q, p), Influence(q)), \quad (1)$$

where  $f()$  is a comprehensive function which will be discussed in our algorithms.

With the above notations, the academic paper recommendation problem can be described as follows. Given a citation-collaboration graph  $G = \langle V_1, V_2, E_1, W_1, E_2, W_2, E_3, W_3 \rangle$  and a paper  $p \in V_1$ , find the top-K ranking papers in the graph. That is, we want to find a set  $S$  satisfies:  $S \subset V_1, |S| = K$ , and  $\forall x \in S, y \in V_1 - S, Rank(x, p) \geq Rank(y, p)$ .

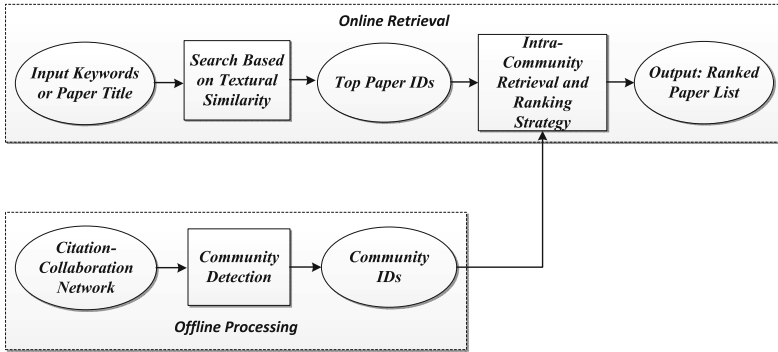


Fig. 2. The framework of paper recommendation system.

## 4 Methodology

To solve the academic paper recommendation problem, we need to search the high rank papers efficiently. A citation-collaboration graph may contains billion of nodes (papers and authors), thus exhausting search in the whole graph is highly inefficient. We propose a community-based strategy to solve the problem efficiently.

Figure 2 shows the framework of our solutions. The proposed method includes two stages: *offline processing* and *online retrieval*. In the offline processing stage, we apply community detection algorithm to divide the citation-collaboration graph into communities. Each community contains about thousands of papers. They are stored in a distributed way, and academic paper recommendation only consider the papers within the same community since they are more close in structure and more related due to their close citation relationships. Community detection can be run offline and can be updated daily or weekly when more papers are included in the system. The online retrieval process includes several steps. First, when a user input the keywords or the title of a paper, we run an initial search algorithm based on textural similarity and got the top-10 paper IDs in the system. Then for each paper ID, we locate its community, and run an intra-community retrieval algorithm and rank the papers according to their structural similarity and influence information. Finally, we sort the papers according to their ranks and output the top-K papers for recommendation.

The detailed algorithms are discussed in the following sections.

## 5 Community Detection

Community detection algorithms in complex networks have been widely studied in the past. However, since the citation-collaboration graph is a heterogeneous network with three different types of edges, the existing community detection strategy cannot be directly applied. To deal with the problem, we first transform

the two-layer heterogeneous graph to a one-dimensional network, and then apply a weighted label propagation algorithm for community detection. The details are as follows.

## 5.1 Graph Transformation

We transform the citation-collaboration graph  $G = \langle V_1, V_2, E_1, W_1, E_2, W_2, E_3, W_3 \rangle$  to a one-dimensional directed weighted graph  $G' = \langle V', E', W' : E' \subseteq V' \times V' \rangle$  using the following methods.

Firstly, we need to turn heterogeneous sets of nodes  $V_1, V_2$  into a unified set. Since our objective is to recommend academic papers, we can only focus on the set of papers  $V_1$  and map the weights of  $V_2$  to the set  $V_1$ . Therefore we let  $V' = V_1$  after graph transformation.

Secondly, we evaluate the relative importance of citation network and co-author network respectively. The weight of each edge in citation network is 1, while the edges in collaboration network have various weights depending on the number of co-publications. To deal with this issue, we introduce a factor  $\gamma$  to represent the relative importance when transforming the weights from the collaboration network to the citation network. Apparently,  $0 < \gamma < 1$  is a turnable parameter and we let  $\gamma = 0.5$  in our system.

We map the co-author relationship in the collaboration graph to the transformed graph as follows. If paper  $p_i$  and  $p_j$  have a common author, then there are bi-directional links between  $p_i$  and  $p_j$ . The weight on the link is calculated by considering both citation relationship and author similarity.

$$w'_{ij} = \begin{cases} 1 + \gamma * AuthorSim(p_i, p_j); & (p_i, p_j) \in E_1 \\ \gamma * AuthorSim(p_i, p_j); & otherwise; \end{cases}$$

After applying the above method, the citation-collaboration graph can be transformed to the new graph  $G' = \langle V', E', W' \rangle$ , where  $V'$  is the set of papers, and  $E'$  and  $W'$  are the new set of edges and weights computed by the transforming method.

## 5.2 Weighted Label Propagation Algorithm

After transforming the two-level complex network into a single dimensional graph, we can apply a community detection algorithm to divide the graph to communities. To achieve community detection in an efficient way, we adopt the linear-time label propagation algorithm (LPA) which is introduced in [12]. However, the original LPA is designed for undirected unweighted graphs, which can not be applied directed in our system. To adapt the LPA to the directed weighted graph as in our case, we proposed a *weighted label propagation algorithm (WLPA)* shown in Algorithm 1.

In the algorithm,  $Neighbor(v)$  is the neighbor set of  $v$  in  $G'$ . The algorithm contains the following steps.

- (1) Initialization: assign each node a different community label.

**Algorithm 1.** Weighted Label Propagation Algorithm**Input:** transformed network  $G' = (V', E', W')$ .**Output:** Community labels for each node in  $G'$ .

---

```

1: initialize nodes with unique labels
2: while labels of nodes are not stable do
3:   for all  $v \in V'$  do
4:     for all  $u \in Neighbor(v)$  do
5:        $l_u = \text{the label of } u$ 
6:        $sum(l_u) = sum(l_u) + w'_{vu}$ 
7:     end for
8:     update the label of  $v$ :  $l_v = argmax_{l_i} \{sum(l_i) \forall i \in Neighbor(v)\}$ 
9:   end for
10: end while

```

---

(2) Iteration: In the iteration process, each node selects the most “adaptable” label in its neighbors and alter its own label to the same label. Unlike the original LPA, the WLPA only consider out-degree neighbors (which means the papers cited by the individual). Then the node consider the labels of its neighbors and the weights, and updates its label to the one with the maximum accumulative weights, which is

$$l_v = argmax_{l_j} \sum_{j \in Neighbor^{l_j}(v)} w'_{vj} \quad (2)$$

where  $Neighbor^{l_j}(v)$  represents the set of neighbors of  $v$  with label  $l_j$ , and the weight of edge between  $v$  and  $j$  is  $w'_{vj}$ .

Repeating this procedure until most of the nodes in graph remain stable labels and don’t change their community labels in the next iterations. According to the analyze of LPA [12], the algorithm converges in constant rounds.

## 6 Online Retrieval and Ranking

When a user input the paper title or some keywords, we first use the textural similarity to perform initial search to obtain the top-K papers that are most relevant to the input keywords. In the system implementation, we use the Lucence engine [17], a search index engine, to automatically abstract keywords from the literature to form index, and to obtain the textural similarity values using the TF-IDF algorithm [18]. Lucence engine has been proved to be very efficient for text-based searching in large-scale distributed database [17].

With the top-K papers from text searching, we can identify the corresponding community ID for each paper based on the community detection result. For each paper in a community, we evaluate its rank according to the network structure information. We introduce two ranking algorithms.



## 6.1 Adaptive Ranking Algorithm

The adaptive ranking algorithm works as follows. Firstly, given the top  $i$ th paper obtained from the textural similarity search, we can get its community ID, and then obtain the set of papers in the same community  $C_i$ . Secondly, we rank each paper in community  $C_i$  by comprehensively considering the metrics of textural similarity, influence and closeness. For an input paper  $p$  and each  $p_j$  in  $C_i$ ,

$$\text{Rank}(p_j, p) = \lambda_1 * \text{TextSim}(p, p_j) + \lambda_2 * \text{Influence}(p_j) + \lambda_3 * \text{Closeness}(p, p_j),$$

where  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are turnable parameters to indicate the importance of each metric in the total rank.

Thirdly, we repeat the above process for all the top-K papers obtained from the textural similarity search, and sort the papers in  $\bigcup_{i=1}^K C_i$  according to their ranking scores. Finally, we output the top-K papers with the highest ranking scores, which are recommended to the user.

## 6.2 Random Walk Algorithm

The other ranking algorithm we proposed is a random walk algorithm. Given a paper represented by a node in the graph, we make several rounds of random walk starting from the node with a fixed walking distance. If a node is reachable for multiple times, it means that it is important in the network structure and can be candidate for recommendation. To consider the closeness of two nodes, the probability of choosing the next waking node should be proportional to the weight on the links. The ranking of a paper can be computed by a comprehensive function of both the textural similarity and the times of visits in the random walks.

At each step of random walks, the walker randomly chooses one of its neighbors to move. The visiting sequence of this walker can be described as a Markov chain. The probability of walking from node  $p_i$  to node  $p_j$  is defined by

$$P(i, j) = \frac{w'_{ij}}{\sum_k w'_{ik}}. \quad (3)$$

The total length of walks is denoted by  $L$ . The longer walking distance, the less relevant of the papers. In our system, we set  $L = 10$ . The random walks repeat for several rounds  $R$  and we use  $\text{VisitTime}(p_j)$  to denote the times that node  $p_j$  being visited. Intuitively, if two papers are close, they will be easy to reach each other, and the visit times should be high.

After random walks, we can compute the ranking of each node  $p_j$  being visited during walks starting from paper  $p$ .

$$\text{Rank}(p_j, p) = \lambda_1 * \text{TextSim}(p, p_j) + \lambda_4 * \text{VisitTime}(p_j), \quad (4)$$

where  $\lambda_1$  and  $\lambda_4$  are turnable parameters to indicate the importance of each metric in the total rank.

After sorting the result according to the ranking scores, the algorithm output the top-K papers for recommendation.

## 7 Performance Evaluation

We implement the proposed paper recommendation algorithms in the academic searching engine and evaluate their performance. The dataset we used is from the Microsoft Academic Graph [2], which contains the meta information of more than 120 million scientific publication records, citation relationships between those publications, as well as authors, institutions, journals, conference venues, and fields of study.

We compare the performance with a baseline algorithm based on the popular PageRank method [19]. Specifically, given an input paper  $p$ , we consider a modified PageRank algorithm to compute the ranking of paper  $p_j$  as follows.

$$\text{Rank}(p_j, p) = \lambda_1 * \text{TextSim}(p, p_j) + \lambda_5 * \text{PageRank}(p_j), \quad (5)$$

where  $\text{PageRank}(p_j)$  is the page rank value obtained from the conventional PageRank algorithm, and  $\lambda_1$  and  $\lambda_5$  are turnable parameters.

The default values of the system parameters are summarized in Table 1. The chosen of such values intends to provide fair weights to the impact factors.

**Table 1.** Default values of the system parameters.

System parameters	$\gamma$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	$L$	$R$
Default value	0.5	1.0	0.5	0.5	0.5	0.5	10	1000

To conduct the experiments, we invite 20 volunteers to help on evaluating the quality of academic paper recommendation results. We randomly choose 50 papers in the system and show the searching results to the volunteers. The ranking algorithms are anonymous and random to the volunteers. The volunteers are required to rank the quality of each paper' recommendation result with a score ranging from 1 to 5. The higher score means the better quality. We collect all the scores on the 50 papers and use them for performance evaluation.

Figure 3 compares the cumulative distribution function (CDF) of the scores of the three algorithms. As shown in the figure, for the page rank algorithm, about 70% results are ranked 1, and there are less than 10% results are ranked higher than score 3. The Adaptive algorithm and Random Walk algorithm perform much better than PageRank, where about 20% results are ranked lower than 2, and about 50% results are higher than 3. The Random Walk algorithm performs the best among all the strategies.

We also compare several performance metrics such as textural similarity, number of citations of the recommended paper, which results are shown in Figs. 4 and 5.

Figure 4 compares CDF of textural similarity of the three algorithms. As shown in the figure, PageRank has the lowest textural similarity, where about 90% results have very small textural similarity lower than 0.05. The Random

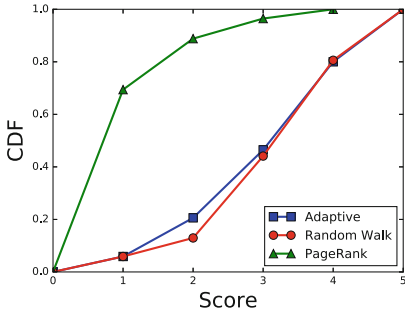


Fig. 3. CDF of ranking scores.

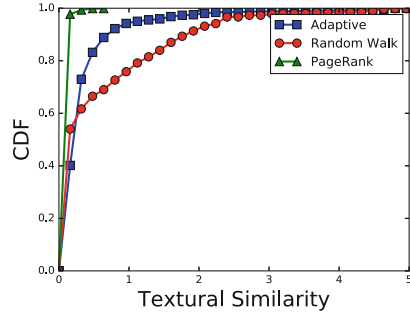


Fig. 4. CDF of textural similarities.

Walk algorithm have 7% results lower than 0.05, and about 30% results higher than 1. The Adaptive algorithm performs in between. Again, the Random Walk algorithm performs the best among the three.

The CDF of citation counts of the algorithms are shown in Fig. 5. The PageRank has the highest citation count, where 90% results have citations higher than 2000. While for Adaptive and Random Walk, more than 90% results have citations less than 1000. It means that PageRank tends to find the high citation papers. However, due to the low textural similarity of PageRank, such high cited papers maybe not relevant to the paper that a user want to search. Although PageRank emphasizes the influence of the paper, it fails to address the similarity of papers, which cannot achieve good performance in practice. The proposed algorithm considers textural similarity, citations and closeness in network structure, which performs better than PageRank in paper recommendation.

To show the time efficiency of the proposed algorithms, we compare the response time of the search engine using different algorithms. Figure 6 shows the CDF of response time of searching 50 papers. According to the figure, 80% searching of Adaptive algorithm are replied within 200–400 ms; 80% searching

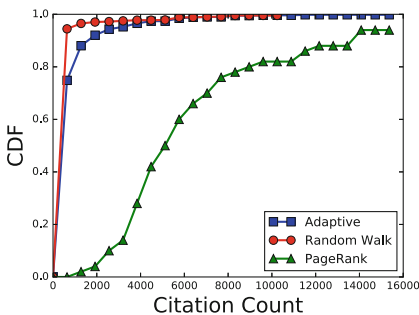


Fig. 5. CDF of citation counts.

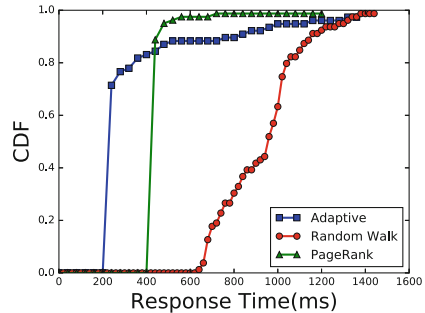


Fig. 6. CDF of response time.

of Random Walk are replied within 400–600 ms, and the PageRank algorithm is within 600–1200 ms most of the time. The proposed algorithms outperforms PageRank in response time and the Adaptive algorithm achieves the highest time efficiency in academic search.

## 8 Conclusion

In this paper, we focused on the issues of paper recommendation for academic search. We showed that by considering the relationships of author collaborations and paper citations can reveal the implicit correlation between publications. Therefore we proposed an academic paper recommendation system based on community detection in citation-collaboration networks. First, we introduced a graph transform method to transform the two-layer citation-collaboration network to a uniform directed graph with nodes representing papers and weights representing the closeness between them. Then, we introduced a weighted label propagation algorithm for community detection to find the most relevant cluster of papers. After dividing the graph into communities, we proposed two ranking algorithms called Adaptive and Random Walk to rank the papers within the same community. We implemented an academic paper recommendation system based on the dataset from Microsoft Academic Graph. The system performance were evaluated based on the volunteers' assessments, which show that the proposed paper recommendation algorithms outperform the conventional search engine algorithm such as PageRank. The efficiency of the proposed algorithms were verified by the evaluation.

**Acknowledgements.** This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 61373128, 91218302, 61321491), the Key Project of Jiangsu Research Program Grant (No. BE2013116), the EU FP7 IRSES MobileCloud Project (Grant No. 612212).

## References

1. Google scholar. <http://scholar.google.com>
2. Microsoft academic. <https://academic.microsoft.com/>
3. Aminer. <https://aminer.org/>
4. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. *Ann. Rev. Sociol.* **27**(1), 415–444 (2001)
5. Filippo, R., Claudio, C.: Defining and identifying communities in networks. *Proc. Nat. Acad. Sci. U.S.A* **101**(9), 2658–2663 (2004)
6. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proc. Nat. Acad. Sci. U.S.A* **99**(12), 7821 (2002)
7. Darrin, E., Bo-June (Paul), H.: Microsoft academic graph. <http://research.microsoft.com/en-us/projects/mag/>
8. Newman, M.E.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **69**(6), 066 133 (2004)
9. Newman, M.E.J.: Detecting community structure in networks. *Eur. Phys. J. B Condens. Matter Complex Syst.* **38**(2), 321–330 (2004)

10. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proc. Nat. Acad. Sci. U.S.A* **105**, 1118 (2008)
11. Bao, F., Licheng, J., Maoguo, G., Haifeng, D.: Memetic algorithm for community detection in networks (2011)
12. Usha Nandini, R., Reka, A., Soundar R T, K.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3), 036106 (2007)
13. Pagerank: U.S. Patent 6 285 999
14. Jon, K.: Authoritative sources in a hyperlinked environment. *J. ACM* **46**(5), 604–632 (1999)
15. Haveliwala, T.H.: Topic-sensitive pagerank. In: *Proceedings of the 11th International Conference on World Wide Web (WWW 2002)*, pp. 517–526. ACM, New York (2002)
16. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Netw.* **1**(3), 215–239 (1978)
17. Hatcher, E., Gospodnetic, O.: Lucene in action (in action series) (2004)
18. Hearst, M.A.: Tilebars: visualization of term distribution information in full text information access. In: *Sigchi Conference on Human Factors in Computing Systems*, pp. 631 – 637 (1995)
19. Page, L.: The pagerank citation ranking: Bringing order to the web. In: *Stanford InfoLab*, pp. 1–14 (1998)