An Analysis of Citation Recommender Systems: Beyond the Obvious

Haofeng Jia and Erik Saule
Dept. of Computer Science, University of North Carolina at Charlotte
{hjia1,esaule}@uncc.edu

***Abstract*—As science advances, the academic community has published millions of research papers. Researchers devote time and effort to search relevant manuscripts when writing a paper or simply to keep up with current research. In this paper, we consider the problem of citation recommendation by extending a set of known-to-be-relevant references. Our analysis shows the degrees of cited papers in the subgraph induced by the citations of a paper, called projection graph, follow a power law distribution. Existing popular methods are only good at finding the long tail papers, the ones that are highly connected to others. In other words, the majority of cited papers are loosely connected in the projection graph but they are not going to be found by existing methods. To address this problem, we propose to combine author, venue and keyword information to interpret the citation behavior behind those loosely connected papers. Results show that different methods are finding cited papers with widely different properties. We suggest multiple recommended lists by different algorithms could satisfy various users for a real citation recommendation system.**

I. Introduction

Scientists around the world have published tens of millions of research papers, and the number of new papers has been increasing with time. For example, according to DBLP [1], computer scientists published 3 times more papers in 2010 than in 2000. At the same time, literature search became an essential task performed daily by thousands of researcher around the world. Finding relevant research works from the gigantic number of published articles has become a nontrivial problem.

Currently, many researchers rely on manual methods, such as keyword search via Google Scholar[1] or Mendeley[2], to discover new research works. However, keyword based searches might not be satisfying for two reasons: firstly, the vocabulary gap between the query and the relevant document might results in poor performance; secondly, a simple string of keywords might not be enough to convey the information needs of researchers. There are many instances where such a keyword query is either over broad, returning many articles that are loosely relevant to what the researcher actual need, or too narrow, filtering many potentially relevant articles out or returning nothing at all [2].

To alleviate the above mentioned problems, many research works proposed citation recommendation algorithms which use a manuscript instead of a set of keywords as query [3], [4], [5], [6], [7]. For example, context-aware citation recommendation is designed to recommend relevant papers for placeholders in the query manuscript based on local contexts [4], [5]. Manuscript based citation recommendation is great to help with the writing process. However, we are interested here in helping the research process which usually comes long before manuscripts are fleshed out. Researchers have devoted efforts on citation recommendation based on a set of seed papers [8], [9], [10], [11], [12]. Most approaches rely on the citation graph to recommend relevant papers, such as collaborative filtering [8] and random walk framework [10]. The different approaches to recommending academic papers have been extensively surveyed by [13].

We consider in this paper the problem of extending a set of known references, which is helpful in recommender system and academic search engine, such as theAdvisor [14]. We show that classic methods (namely, PaperRank and Collaborative Filtering) perform reasonably well, but have an inherent bias. Because they base their decision on citation patterns, they tend to only find papers that have many links to the set of known references, a set of papers that are obvious. Unfortunately, less than half of the references of a paper are connected to more than two other references. This causes the algorithms to ignore lightly connected papers despite being half of the references in practice.

We propose to use metadata information, such as authorship and textual information, to identify the non-obvious connections between papers. We design two types of algorithms. One type extends PaperRank with some metadata; C+K extends PaperRank by adding keyword nodes in the graphs to enable paths between papers with similar keywords. And one that only uses metadata; logAVK directly scores candidate papers based on the similarity of their metadata to the query papers. Our experiments show that the methods that extend PaperRank can improve the quality of the recommendation. It also shows that logAVK provides a different perspective on the queries, despite it does not score as well in various quality metrics as other algorithms.

The rest of the paper is organized as follows: we introduce related work in Section II. In Section III, we define the citation recommendation problem and present existing methods. Based on the analysis of Section IV, we propose to combine metadata information to enhance the performance on loosely connected papers in Section V. Section VI argues the use of the different algorithms in a practical system. Finally, Section VII we conclude our work and discuss the usefulness in real systems.

II. Related Work

Various approaches have been proposed for citation recommendation.

[1] https://scholar.google.com/
[2] https://www.mendeley.com/

Seed papers based citation recommendation. Given a "basket" of citations, McNee et al. [8] explore the use of Collaborative Filtering (CF) to recommend papers that would be suitable additional references for a target research paper. They create a ratings matrix where citing papers correspond to users and citations correspond to items. The experiments show CF could generate high quality recommendations. As a follow-up, Torres et al. [9] describe and test different techniques for combining Collaborative Filtering and Content-Based Filtering. A user study shows many of CF-CBF hybrid recommender algorithms can generate research paper recommendations that users were happy to receive. However, offline experiments show those hybrid algorithms did not perform well. In their opinion, the sequential nature of these hybrid algorithms: the second module is only able to make recommendations seeded by the results of the first module. To address this problem, Ekstrand et al. [15] propose to fuse the two steps by running a CF and a CBF recommender in parallel and blending the resulting ranked lists. The first items on the combined recommendation list are those items which appeared on both lists, ordered by the sum of their ranks. Surprisingly, Collaborative Filtering outperforms all hybrid algorithms in their experiments.

Gori et al. [10] devised a random walk based method, to recommend papers according to a small set of user selected relevant articles. Küçüktunç et al. designed a personalized paper recommendation service, called theAdvisor[3] [12], [14], which allows a user to specify her search toward recent developments or traditional papers using a direction-aware random walk with restart algorithm [16]. The recommended papers returned by theAdvisor are diversified by parameterized relaxed local maxima [17]. Küçüktunç et al. proposed sparse matrix ordering and partitioning techniques to accelerate citation such recommendation algorithms [18].

Caragea et al. [11] addressed the problem of citation recommendation using singular value decomposition on the adjacency matrix associated with the citation graph to construct a latent semantic space: a lower-dimensional space where correlated papers can be more easily identified. Their experiments on Citeseer digital library show this approach achieves significant success compared with Collaborative Filtering methods. Wang et al. [19] proposes to include textual information to build an topic model of the papers and adds an additional latent variable to distinguish between the focus of a paper and the area it just talks about.

A typical related paper search scenario is that a user starts with a seed of one or more papers, by reading the available text and searching related cited references. Sofia is a system that automates this recursive process [20].

The approach proposed by [2] returns a set of relevant articles by optimizing an objective function based on a fine-grained notion of influence between documents. El-Arini et al. also claim that, for paper recommendation, defining a query as a small set of known-to-be-relevant papers is better than a string of keywords [2].

Manuscript based citation recommendation. Stohman et al. [3] examined the effectiveness of various text-based and citation-based features on citation recommendation, they find that neither text-based nor citation-based features performed very well in isolation, while text similarity alone achieves a surprisingly poor performance at this task. He et al. [4] considered the problem of recommending citations for placeholder in query manuscripts and a proposed non-parametric probabilistic model to measure the relevance between a citation context and a candidate citation. To reduce the burden on users, [5] proposed different models for automatically finding citation contexts in an unlabeled query manuscript.

Recently, citation recommendation from heterogeneous network mining perspective has attracted more attention. Besides papers, metadata such as authors or keywords are also considered as entities in the graph schema. Two entities can be connected via different paths, called meta-paths, which usually carry different semantic meanings. Many work build discriminative models for citation prediction and recommendation based on meta-paths [21], [22], [23], [24].

The vocabulary used in the citation context and in the content of papers are usually quite different. To address this problem, some works propose to use translation model, which can bridge the gap between two heterogeneous languages [6], [7]. Based on previous work [4], [5], [7], Huang et al. built a citation recommendation system called RefSeer[4] [25] which perform both topic-based global recommendations and citation-context-based local recommendations.

Based on the hypothesis that an author's published works constitute a clean signal of the latent interests of a researcher, [26] examined the effect of modeling a researcher's past works in recommending papers. Specifically, they first construct a user profile based on her/his recent works, then rank candidate papers according to the content similarity between the candidate and the user profile. Furthermore, in order to achieve a better representation of candidate paper, [27] exploit potential citation papers through the use of collaborative filtering.

III. Citation Recommendation

A. Data Preparation

To obtain a clean and comprehensive academic data set, we match Microsoft Academic Graph[5] [28], CiteSeerX[6] and DBLP[7] [1] datasets for their complementary advantages and derive a corpus of Computer Science papers. On one hand, Microsoft Academic Graph contains abundant information from various disciplines but it is fairly noisy: some important attributes are missing or even wrong. In contrast, the records in DBLP are much more reliable although it does not contain citation information. So we first merge MAG and DBLP records through DOI and titles to get an academic citation graph (within the scope of Computer Science) with rather clean metadata.

On the other hand, CiteSeerX dataset indexes 2 million papers and provides full-texts in PDF format which neither MAG or DBLP contains. We merge CiteSeerX and DBLP records through titles and refine the result with other metadata, like published year.

This data set gives us for each paper the name of the authors, the venue of publication, the title of the paper, full text (for about a fifth of the papers), and citation information.

[3] http://theadvisor.osu.edu/

[4] http://refseer.ist.psu.edu/

[5] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

[6] http://citeseerx.ist.psu.edu/

[7] http://dblp.uni-trier.de/xml/

TABLE I: Data Statistics

Attribute	Number
Papers	2,035,246
Citations	12,439,090
Papers with text	374,999
Keywords	195,989
P-K Edges	14,779,751

Attribute	Number
Authors	1,208,641
P-A Edges	5,977,884
Venues	9,777
P-V Edges	2,035,246

We derived keywords using KP-Miner [29] for those with full text and using non-stop words from titles for others. High level statistics of this dataset is given in Table I.

B. Problem Statement

Let $G = (V, E)$ be the citation graph, with n papers $V = \{v_1, \ldots, v_n\}$. In G, each edge $e \in E$ represents a citation relationship between two papers. We use $Ref(v)$ and $Cit(v)$ to denote the reference set of and citation set to v, respectively. And $Adj(v)$ is used to denote the union of $Ref(v)$ and $Cit(v)$.

In this paper, we focus on citation recommendation problem assuming that researchers already know a set of relevant papers. Therefore, the task can be formalized as:

Citation Recommendation. Given a set of seed papers S, return a list of papers ranked by relevance to the ones in S.

C. Algorithms

a) CoCitation [30]: The number of cocitations is often used to measure the relevance between two papers. In the citation recommendation scenario, cocitation ranks a candidate paper according to the sum of the times it was cocited with papers in the seed set.

$$R(p) = \sum_{s \in S} \sum_{v \text{ for } s, p \in Cit(v)} 1$$

b) CoCoupling [30]: CoCoupling is a complementary metric of cocitation. It counts the number of times that two papers cite a same paper. Here, we use cocoupling to measure the relevance between the candidate paper and seed papers according to the following formula:

$$R(p) = \sum_{s \in S} \sum_{v \text{ for } s, p \in Ref(v)} 1$$

c) PaperRank [10] (PR): PaperRank is a biased random walk proposed to recommend papers based on citation graph. In particular, the restarts from any paper will be distributed to only the seed papers. PR assumes a random walker in paper v continues to a neighbor with a damping factor d, and with probability $(1 - d)$ it restarts at one of the seed papers in S. The edges are followed proportionally to the weight of that edge w_{ji} which is often set to 1, but can be set to the number of time i is referenced by j.

$$R(v_i) = (1 - d)\frac{1}{S} + d \times \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} R(v_j)$$

d) Collaborative Filtering [8] (CF): has been proven to be an effective idea for most recommendation problems. For citation recommendation, a ratings matrix is built using the adjacent matrix of citation graph, where citing papers correspond to users and citations correspond to items. A pseudo target paper that cites all seed papers is added to the matrix. CF computes the k neighborhoods that are top k similar papers to the target paper. Then each citation in neighbors is summed up with the count weighted by the similarity score.

TABLE II: Global Performance

Method	Recall@10	Recall@20	Recall@50
PaperRank	0.234413	0.326096	0.471510
CF	0.191736	0.266961	0.391736
CoCitation	0.192626	0.267617	0.392197
CoCoupling	0.055778	0.088216	0.146737

IV. A First Evaluation

In order to simulate the typical usecase where a researcher is writing a paper and tries to find some more references, we design the random-hide experiment. First of all, a query paper q with 20 to 200 references and published between 2005 to 2010 is randomly (uniformly) selected from the dataset. We then remove the query paper q and all papers published after q from the citation graph to simulate the time when the query paper was being written. Instead of using hide-one strategy [8], [9], we randomly hide 10% of the references as hidden set. This set of hidden paper is used as ground truth to recommend. The remaining (18 and 180 depending on q) papers are used as the set of seed papers.

Finally, to evaluate the effectiveness of recommendation algorithm, we use *recall@k*, the ratio of hidden papers appearing in top k of the recommended list. Table II shows the results of popular methods on average recall for 2,500 independent randomly selected queries. We call these scores global performance, as we will analyze the common features of the hidden papers found by those methods to reveal the bias behind the algorithms.

To analyze the performance of the algorithms, we investigate the local structure of the citation graph. The citation projection graph of a paper p is the graph induced by the papers cited by p [31]. For a query paper, it is the graph where the vertex set is composed of the seed papers and the hidden papers, and the edge set is composed of the citations between these vertices. The citation projection graph focuses on the cited papers and the relationships among them; it is known to help understanding the local pattern in the citation graph [31].

We investigated the relations between various properties of the projection graph and whether hidden papers were found or not. We identified that the degree of the hidden papers in the projection graph, we call proj-degree, is a good indicator of whether the hidden paper will be discovered or not. We computed the average recall@10 scores on hidden papers grouped by proj-degree and reported these numbers in Figure 1. Popular graph based methods are quite good at finding hidden papers that are highly connected in the citation projection graph. But these methods achieve poor performance on loosely connected ones. Unfortunately, over 50% of the hidden papers have a proj-degree of 2 or less. The distribution of proj-degree is shown in Figure 2.

V. Finding Loosely Connected Papers

A. Are these papers random citation?

The analysis of the last section shows that popular methods are good at finding papers that are highly connected in citation projection graph. But they perform poorly on papers that are not well connected in the citation projection graph despite they are the majority. Therefore, we focus our analysis on loosely connected papers.

The key question is why do authors cites these papers?

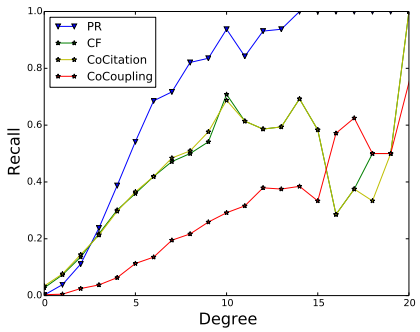


Fig. 1: Recall by proj-degree (top 10)

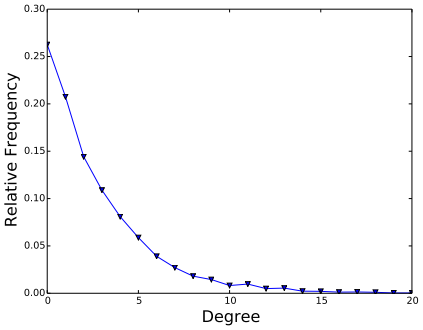


Fig. 2: Distribution of proj-degree (degree in the citation projection graph) of hidden papers.

According to [31], some papers create random reference across various fields. This might sound reasonable to explain the fact that these reference are loosely connected in the citation projection graph. However, as Figure 2 shows, about 50% of cited papers have one or no link to others. Therefore, we believe they must share some common patterns or features with others cocited papers that are not apparent in the citation graph. We expect other features such as authors, venue, or keywords, convey helpful information.

A preliminary analysis of the medata of loosely connected papers shows that about 46% of the papers connected to two of the seeds or less share at least one common author with at least one of the seed papers. 60% of the loosely connected papers appeared in the venue of one of the seed paper. And 95% of the loosely connected paper shared at least one keyword with one of the seed paper. This indicates that the citations are not random citations; but authors chose to cite them for reasons that are not clearly explained by the citation graph.

B. Algorithms using Metadata

Based on above analysis, we propose two random walk based methods to examine the ability of metadata for identifying loosely connected papers. The first one is based on the metadata themselves, and we combine the metadata and citation graph in the second framework.

1) LOGAVK: In order to compute the similarity between one paper to a set of other papers, we build attribute graphs for author, venue and keyword respectively. Let us take author as example, we first define an undirected weighted graph of authors where an edge represents the number of papers two authors have written together. Then we normalize the adjacent matrix of this graph as M^{AA}, where A is the set of authors. Once the graph is constructed, we can measure the similarity between a candidate author and the authors of seed papers by random walk as follows:

$$R^A = \begin{cases} \alpha M^{AA} R^A + (1-\alpha)\frac{1}{S} & \text{For authors of seed papers} \\ \alpha M^{AA} R^A & \text{otherwise} \end{cases}$$

The keyword graph M^{KK} and venue graph M^{VV} are constructed and the similarity score R^V and R^K are computed in the same way. LogAVK recommends the loosely connected papers according to the summation of the similarity scores of authors, venue and keywords with corresponding seed papers.

$$Score_{LogAVK} = logR^A + logR^V + logR^K$$

2) Combining Citation graph and Metadata: Aiming to combine the citation information and metadata information, we build bipartite graphs with two kinds of nodes: papers and metadata. A random walk algorithm passes information back and forth between the papers and the metadata. Taking the paper-author graph as an example, the vector of paper scores is denoted by R^P and the vector of author scores is denoted by R^A. The scores of authors is computed by:

$$R^A = M^{AP} R^P$$

which means an author score is collected from the papers she published. Some of the paper scores are transferred between papers within the citation graph:

$$R_1^P = M^{PP} R^P$$

And a paper also collects scores from its authors:

$$R_2^P = M^{PA} R^A$$

A paper in the seed set S also receives scores by random jumping from others.

$$R_3^P = \frac{1}{S}$$

The final score of a paper is the weighted sum of above parts.

$$R_A^P = \begin{cases} \alpha R_1^P + \beta R_2^P + (1-\alpha-\beta) R_3^P & \text{for seed papers} \\ \alpha R_1^P + \beta R_2^P & \text{otherwise} \end{cases}$$

where α (β. resp.) is the fraction of the rank following a citation edge (an author edge, resp.). In the experiments, we set $\alpha = .65$, $\beta = .2$.

We will refer to any method that combines the citation information and a metadata in this manner as C+X. In particular, C+A will denote combining citation and authorship; C+K will denote combining citation and keyword; and C+V will denote combining citation and venue.

C. Evaluation

a) General performance: We evaluate the general effectiveness of the recommendation algorithms using *recall@k*, the ratio of hidden papers appearing in top k of the recommended list. Table III shows the results of popular methods and the methods on average recall for 2,500 independent queries.

The results show that the C+X methods do not perform quite as well as PaperRank; while the performance of logAVK is lower than that of PaperRank by a factor of about 4.

TABLE III: Global Performance

Method	Recall@10	Recall@20	Recall@50
PaperRank	0.234413	0.326096	0.471510
CF	0.191736	0.266961	0.391736
C+A	0.230617	0.318206	0.463204
C+V	0.230531	0.323125	0.461898
C+K	0.231308	0.315485	0.461507
LogAVK	0.053934	0.084001	0.129175

TABLE IV: Performance by proj-degree: Recall@10

Method	$\delta = 0$	$\delta = 1$	$\delta = 2$	$\delta = 3$	$\delta = 4$	$\delta = 5$
Cocitation	0.10465	**0.20760**	0.46879	0.73310	0.88773	0.94630
Cocoupling	0.01723	0.03577	0.11962	0.25298	0.46489	0.64429
Co. Filtering	0.09914	0.20051	0.47150	0.73821	0.89120	0.94630
PaperRank	0.12172	0.20284	0.50463	**0.76831**	0.91358	0.96979
C+A	0.11193	0.20719	**0.51515**	0.76490	0.91319	**0.97147**
C+V	0.11544	0.18023	0.49932	0.76661	**0.92168**	**0.97147**
C+K	**0.14160**	0.17829	0.50260	0.76008	0.91242	**0.97147**
logAVK	0.02394	0.10287	0.30427	0.55820	0.80671	0.91778

b) Performance by proj-degree: In order to evaluate the ability to recommend papers with a particular degree in the citation projection graphs, we design the second experiment. We define *recall@*k for $\delta = \Delta$ as the ratio of hidden papers with proj-degree d to seeds papers appearing in top k of the recommended list, where only the papers with proj-degree Δ to seeds papers are considered as candidates[8]. The results are shown in Table IV.

For particular values of proj-degree, the combined methods (C+X) outperform current methods. One can easily see that most methods perform well on high proj-degrees. Indeed, there are few vertices that are very connected with the seed papers. So any reasonable algorithm will find most of them. It is on lower proj-degrees (0, 1, and 2) that the algorithms start finding less than 50% of the hidden papers.

Figure 3 shows the evolution of the recall when the number of returned papers varies for three definitions of low proj-degree ($\delta = 0$, $\delta \leq 1$, $\delta \leq 2$). The performance of the algorithms for $\delta \leq 1$ and $\delta \leq 2$ are similar: all graph based methods perform about the same (except cocoupling). logAVK performs significantly worse. For completely disconnect papers ($\delta = 0$), the graph based algorithms exhibit more difference. And in particular, C+K performs better than all other tested algorithms, besting PaperRank by .02. This indicates that metadata help finding loosely connected paper.

VI. On the Usefulness of Different Algorithms

A. Difference between methods

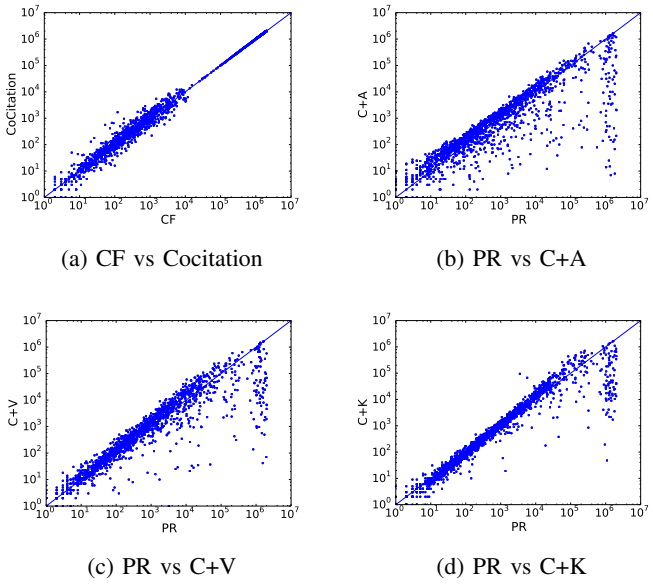
Looking at recall numbers gives a single perspective on the usefulness of the methods. Recall numbers tell us how the algorithms perform on some particular test. While informative to pick a single "best" algorithm, a user wants to explore a dataset and see it through different lenses.

Table V allows us to understand how similar the sets recommended by the algorithms are for loosely connected papers. The diagonal shows the number of hidden papers that were found in the top-10 by a particular algorithm, while an off diagonal entries shows the number of paper found by the algorithm of the row and that were not found by the algorithm on the column. For instance, Cocitation recommended correctly 408 papers but only 393 of those were not correctly identified by CoCoupling.

[8]We call it property proj-degree for simplicity. Indeed the method would need to know which are the hidden paper to do the filtering on proj-degree. We mean degree to the seed, which differs from the real proj-degree by the number of connections to the unknown hidden.

TABLE V: Differences between the top-10 sets ($\delta \leq 2$)

	CoCit	CoCoup	CF	PR	C+A	C+V	C+K	logAVK
CoCit	**408**	393	45	245	284	289	293	389
CoCoup	62	**77**	60	57	64	66	65	75
CF	33	379	**396**	229	268	274	278	378
PR	253	396	249	**416**	83	79	84	396
C+A	259	370	255	50	**383**	58	62	362
C+V	253	361	250	35	47	**372**	17	356
C+K	256	359	253	39	50	16	**371**	356
logAVK	94	111	95	93	92	97	98	**113**



(a) CF vs Cocitation (b) PR vs C+A

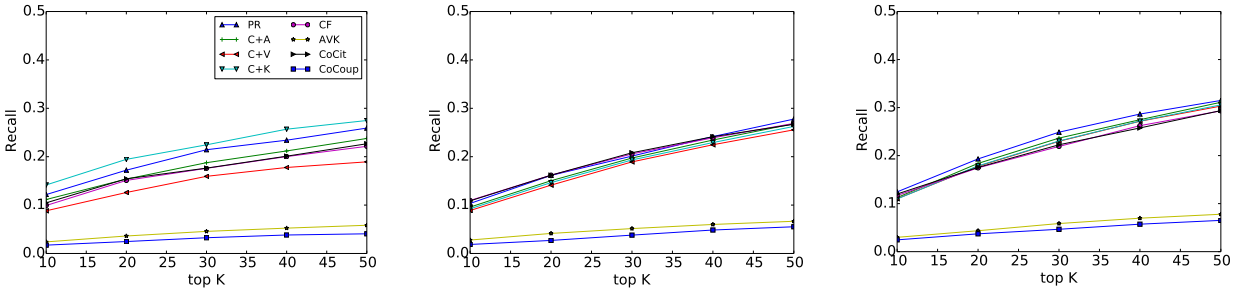
(c) PR vs C+V (d) PR vs C+K

Fig. 4: Rank of hidden papers for $\delta = 0$ (high correlation)

This table allows us to understand that PaperRank, C+A, C+K, and C+V essentially identify the same papers. Indeed each set is composed of about 400 papers, but the difference between these sets is smaller than 100 papers and often smaller than 50 papers. Similarly, Cocitation and Collaborative Filtering both find about 400 papers, but only about 40 of these papers are actually different.

The similarity between these sets is explained by Figure 4 that shows a scatter plot of the ranks of hidden papers in the different algorithms. The rank of hidden papers in the list of Collaborative Filtering and Cocitation are highly correlated. This is not particularly surprising provided Collaborative Filtering and Cocitation are using the same principles with a different weighting function. In other words, Collaborative Filtering and Cocitation are essentially redundant algorithms.

The relations of C+X with PaperRank are somewhat different. There are definitely a strong correlations between these methods, but some papers see a large difference in ranks between the two methods. For instance, two hidden papers were ranked around 1-millionth by PaperRank but was ranked top-10 by C+A. Note also that only few hidden paper see their



(a) $\delta = 0$ (b) $\delta \leq 1$ (c) $\delta \leq 2$

Fig. 3: Performance Comparison for low degree

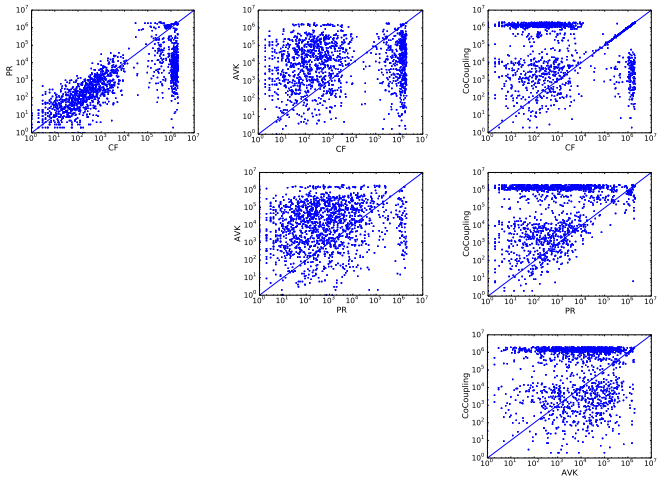


Fig. 5: Rank of hidden papers for $\delta = 0$ (low correlation)

rank being significantly degraded by the addition of an other features (few papers are in the top left corner). This indicates that the algorithms are mostly redundant, but they are using different richer features. As such a better way of using these features could certainly be designed.

Figure 5 shows the correlation of ranks between the remaining algorithms. C+A, C+V, C+K, and Cocitation are not included because of their high correlation with either PaperRank or CF.

The rank comparison of Collaborative Filtering and Cocoupling reveals an interesting structure. Notice that there are some hidden papers with highly correlated with ranks over 10^5. Digging manually in the data show that these hidden papers are not cocited with a seed paper nor are they cociting a common paper with a seed paper. Obviously these papers can not be found by either method. This phenomena explains the denser region of that scatter plots with rank over 10^5 for Collaborative Filtering and CoCoupling.

Collaborative Filtering and PaperRank show some correlation on the papers of rank less than 10^4, though the papers that are not cocited with a seed paper are essentially randomly ordered by Collaborative Filtering.

Cocoupling does not appear to be an interesting algorithm in our test. Indeed, Cocoupling mostly worsens the rank of hidden papers compared to PaperRank (the hidden papers are mostly located in the upper left region).

The logAVK method does not correlate with any other method, nor does it seem to mostly worsen the performance of the paper nor improve them compared to another method. logAVK does provide a completely different perspective on the data than the other algorithms. This is not particularly surprising since it is the only method that does not consider the citation information.

B. Peeking into the Future

The current way of estimating the quality of a paper relies on identifying the papers that were hidden from the list of references of a particular paper. That experiment assumes that the author of each paper is a data point in the ground truth. But authors are imperfect and may not have known some papers. Rather than using a single paper to evaluate the quality of a recommendation, we suggest to use all the future publications.

To quantify the relevance of a recommendation, we define three metrics to explore different aspects of the problem.

c) Relevance-r: For each pair of papers $< i, j >$, where i is a recommended paper and j is a seed paper, we define co-cited probability as:

$$PrCo(i,j) = \frac{|C_{i,j}|}{|C_i|}$$

where $C_{i,j}$ denotes papers citing both i and j in the future and C_i denotes papers citing i in the future. Then, the relevance of a recommended paper to the seed papers is:

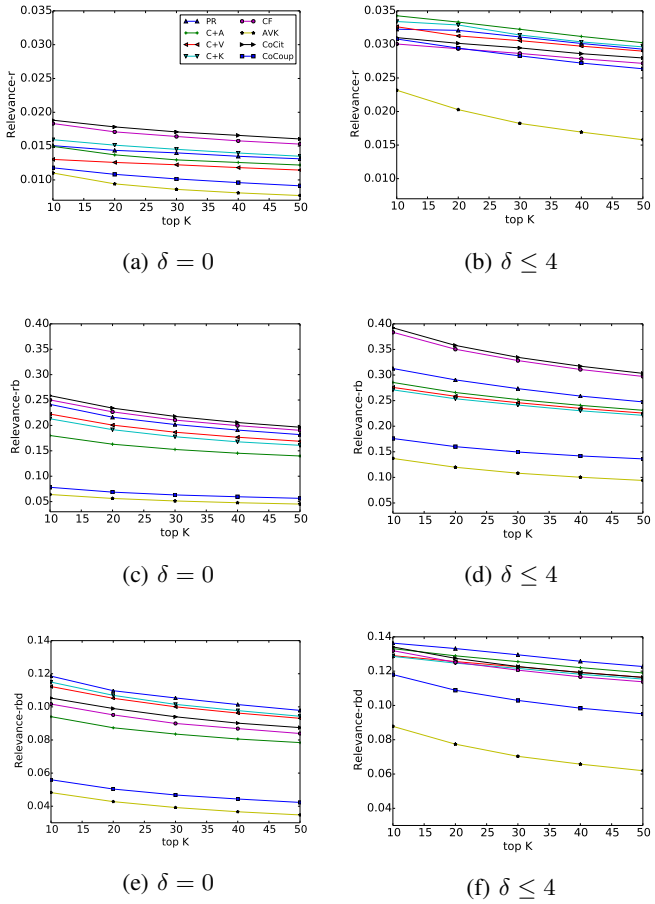
$$Relevance(i) = \frac{\sum_{j \in S} PrCo(i,j)}{|S|}$$

Now we can evaluate the quality of a citation recommendation algorithm by the average relevance for top K results:

$$Relevance@K = \frac{\sum_{i \in topK} Relevance(i)}{K}$$

d) Relevance-rb: The relevance-r between a recommended paper and seed papers could be biased by a few frequently co-cited pairs. To address this problem, we propose a binary version of co-cited probability that just consider about whether there is a paper citing both i and j in the future.

$$PrCo(i,j) = \begin{cases} 1 & \exists C_{i,j} \text{ in the future} \\ 0 & \text{otherwise} \end{cases}$$



(a) $\delta = 0$ (b) $\delta \leq 4$

(c) $\delta = 0$ (d) $\delta \leq 4$

(e) $\delta = 0$ (f) $\delta \leq 4$

Fig. 6: Relevance

e) Relevance-rbd: Note that we are actually interested not only in making good recommendation, but also in making links between papers that were not previously seen as relevant. This version of Relevance only considers the cocitation of a seed-recommended pairs that were not previously cocited.

$$PrCo(i,j) = \begin{cases} 1 & \exists C_{i,j} \text{ in the future and not in the past} \\ 0 & \text{otherwise} \end{cases}$$

We computed the three relevance metrics on the same instances of the problem we run before. We report the results of that experiment in Figure 6.

Not surprisingly, the relevance decreases when the number of returned papers increases. But the relevance does not decrease as fast as one could expect. For instance on $\delta = 0$, the relevance-r of algorithm C+V decreases from .013 to .011 when k goes from 10 to 50. It means that 1.3% of the future citation to the top-10 papers recommended by C+V were in papers that also cited a seed paper; while the relevance-r of top-50 was 1.1%. In other words, the 50th paper recommended by C+V is not much more irrelevant than the 10th.

Not surprisingly, current cocitations is a good predictor of future cocitation: The Collaborative Filtering and Cocitation algorithm perform usually best on the relevance-r and relevance-rb metrics. Though when looking at relevance-rbd that removes the citations that were already known in the present, Collaborative Filtering and Cocitation no longer are the better algorithms. PaperRank is the algorithm that find the most relevant relations that were not known before.

TABLE VI: Upper bound for $\delta = 0$

Metric	top-10	top-20	top-30	top-40	top-50
Relevance_r	0.286093	0.205920	0.170241	0.148972	0.134334
Relevance_rb	0.880969	0.702677	0.590164	0.520564	0.473333
Relevance_rbd	0.778027	0.605512	0.505168	0.443790	0.402499

TABLE VII: Upper bound for $\delta \leq 4$

Metric	top-10	top-20	top-30	top-40	top-50
Relevance_r	0.368085	0.272176	0.225938	0.197591	0.178001
Relevance_rb	0.998309	0.975889	0.879426	0.787065	0.717206
Relevance_rbd	0.868585	0.768658	0.674373	0.600422	0.545786

It is also interesting to see that over 20% of the recommended-seed pairs for PaperRank will be cited in the future and half of these pairs were not known at the time. This suggests that the algorithms we test are actually much more helpful in practice than simple recall tests suggest. The logAVK method also performs interestingly. About 6% of the recommended-seed pairs will be cited in the future (at top-10) and most of them have not been cited before (5% at top-10).

We computed upper bounds on the relevance metrics to quantify how good the different algorithms are. Indeed, we can use the knowledge of the future to easily compute for each query the relevance of each paper and greedily pick the k papers of highest relevance. We report the upper bound on best relevance for $\delta = 0$ in Table VI and for $\delta \leq 4$ in Table VII. The upper bounds are much higher than the relevance of the algorithms: a factor of 10 on relevance-r, 4 on relevance-rb, and 5 on relevance-rbd. This indicates that there is a significant room for improvement in our paper recommendation tasks: there are better set of papers that will be cocited with the seed papers than the methods are recommending.

C. Implications for a practical system?

We evaluated many algorithms, namely PaperRank, Collaborative Filtering, Cocitation, Cocoupling, C+A, C+V, C+K, and LogAVK. The evaluation was performed across different tests, metrics, and by looking at different slices of the solution space. We present here a summary of the discussion with a focus on selecting algorithms for inclusion in a practical system.

Cocitation and Collaborative Filtering are variations of the same algorithm and their performance are hard to distinguish. (See correlation in Figure 4 and the difference in recommendation in Table V). There is no point in including both algorithms in a system: we will pick Collaborative Filtering.

Cocoupling is often one of the worst algorithm and is essentially worse than PaperRank. (See correlation plot in Figure 5). As such, we do not believe it makes sense to include Cocoupling if any variants of PaperRank were to be included.

The C+V, C+A, C+K algorithms are somewhat correlated to PaperRank but they exhibit improvement for many cases (see Figure 4). C+K has the highest recall on the $\delta = 0$ study case (see Figure 3), and C+A and C+K showed the highest relevance-r in the $\delta \leq 4$ case (see Figure 6). However, we believe only one of these methods should be included in practice, but certainly more work in integrating metadata in the recommendation is necessary.

The logAVK algorithm provides a much lower recall than the other algorithm (See Figure 3 for example). However, we believe it could be of some interest to discover loosely

connected papers. Indeed, it returns papers that are very different from the other methods (See Table V) while having a relevance that is within a factor of 2 or 3 of the other algorithms (see Figure 6 for $\delta = 0$). We believe that LogAVK could provide a view of the problem that is complementary to the one provided by the citation based methods.

VII. Conclusion

This manuscript investigates the problem of recommending a set of papers to extend a query composed of a set of known paper. This problem is common in academic recommender systems and academic search engines. The two most common citation recommendation algorithms, PaperRank and Collaborative Filtering, do not uniformly discover relevant papers; they mostly find a set of papers that are highly connected to the query by citations. Unfortunately, real-world citation patterns are not as obvious to find since about 50% cocited papers do not have a direct connection. The key to improving the quality of an academic recommender system lies in identifying those loosely connected, yet relevant, papers. While we consider the problem of identifying highly connected papers essentially solved by the existing methods.

We provided two ways of discovering citations that use the metadata of the papers rather than their citation patterns, LogAVK that only uses the metadata and the C+X algorithm which combine the citation pattern and the metadata. The C+K and C+A algorithms are promising in retrieving papers that are loosely connected to the query. Despite logAVK is about 3 times less relevant than PaperRank, it identifies papers that are known to be important and which are likely to be unknown to the user and the community.

Using a single test/metric to qualify algorithms provides an incomplete picture of how good an algorithm is. We believe that the proposed relevance metrics provide additional insights in the quality and desirability of algorithms.

Future works will focus on building new models for integrating metadata inside a random walk framework to connect better similar papers that are not connected by citations. Currently existing methods require to return a large number of papers to achieve desired recall. Therefore, there is a need in presenting a set of paper to a user in a structured non-list format so that the user can easily navigate the recommendation and identify the papers that appear more relevant.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1652442.

References

[1] M. Ley, "DBLP - some lessons learned," *PVLDB*, vol. 2, no. 2, pp. 1493–1500, 2009.

[2] K. El-Arini and C. Guestrin, "Beyond keyword search: discovering relevant scientific literature," in *Proc of KDD*, 2011, pp. 439–447.

[3] T. Strohman, W. B. Croft, and D. Jensen, "Recommending citations for academic papers," in *Proc. of SIGIR*, 2007, pp. 705–706.

[4] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles, "Context-aware citation recommendation," in *Proc. of WWW*, 2010, pp. 421–430.

[5] Q. He, D. Kifer, J. Pei, P. Mitra, and C. L. Giles, "Citation recommendation without author supervision," in *Proc. of WSDM*, 2011, pp. 755–764.

[6] Y. Lu, J. He, D. Shan, and H. Yan, "Recommending citations with translation model," in *Proc. of CIKM*, 2011, pp. 2017–2020.

[7] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach, "Recommending citations: translating papers into references," in *Proc. of CIKM*, 2012, pp. 1910–1914.

[8] 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 *Proc. of CSCW*, 2002, pp. 116–125.

[9] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl, "Enhancing digital libraries with techlens+," in *Proc. of JCDL*, 2004, pp. 228–236.

[10] M. Gori and A. Pucci, "Research paper recommender systems: A random-walk based approach," in *Proc. of Web Intelligence*, 2006, pp. 778–781.

[11] C. Caragea, A. Silvescu, P. Mitra, and C. L. Giles, "Can't see the forest for the trees?: a citation recommendation system," in *Proc. of JCDL*, 2013, pp. 111–114.

[12] O. Küçüktunç, E. Saule, K. Kaya, and Ü. V. Çatalyürek, "Towards a personalized, scalable, and exploratory academic recommendation service," in *Proc. of ASONAM*, 2013.

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

[14] O. Küçüktunç, E. Saule, K. Kaya, and Ü. V. Çatalyürek, "TheAdvisor: A webservice for academic recommendation," in *Proc. of JCDL*, 2013, p. 2.

[15] M. D. Ekstrand, P. Kannan, J. A. Stemper, J. T. Butler, J. A. Konstan, and J. T. Riedl, "Automatically building research reading lists," in *Proc. of RecSys*, 2010, pp. 159–166.

[16] O. Küçüktunç, E. Saule, K. Kaya, and Ü. V. Çatalyürek, "Direction awareness in citation recommendation," in *Proc. of DBRank*, 2012, p. 6.

[17] ——, "Diversified recommendation on graphs: Pitfalls, measures, and algorithms," in *Proc. of WWW*, 2013.

[18] O. Küçüktunç, K. Kaya, E. Saule, and Ü. V. Çatalyürek, "Fast recommendation on bibliographic networks," in *Proc. of ASONAM*, 2012.

[19] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 448–456. [Online]. Available: http://doi.acm.org/10.1145/2020408.2020480

[20] B. Golshan, T. Lappas, and E. Terzi, "Sofia search: a tool for automating related-work search," in *Proc. of SIGMOD*, 2012, pp. 621–624.

[21] X. Yu, Q. Gu, M. Zhou, and J. Han, "Citation prediction in heterogeneous bibliographic networks," in *Proc. of SDM*, 2012, pp. 1119–1130.

[22] X. Liu, Y. Yu, C. Guo, Y. Sun, and L. Gao, "Full-text based context-rich heterogeneous network mining approach for citation recommendation," in *Proc. of JCDL*, 2014, pp. 361–370.

[23] X. Liu, Y. Yu, C. Guo, and Y. Sun, "Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation," in *Proc. of CIKM*, 2014, pp. 121–130.

[24] X. Ren, J. Liu, X. Yu, U. Khandelwal, Q. Gu, L. Wang, and J. Han, "Cluscite: Effective citation recommendation by information network-based clustering," in *Proc. of KDD*, 2014, pp. 821–830.

[25] W. Huang, Z. Wu, P. Mitra, and C. L. Giles, "Refseer: A citation recommendation system," in *Proc. of JCDL*, 2014, pp. 371–374.

[26] K. Sugiyama and M.-Y. Kan, "Scholarly paper recommendation via user's recent research interests," in *Proc. of JCDL*, 2010, pp. 29–38.

[27] ——, "Exploiting potential citation papers in scholarly paper recommendation," in *Proc. of JCDL*, 2013, pp. 153–162.

[28] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-j. P. Hsu, and K. Wang, "An overview of microsoft academic service (mas) and applications," in *Proc. of WWW*, 2015, pp. 243–246.

[29] S. R. El-Beltagy and A. Rafea, "Kp-miner: Participation in semeval-2," in *Proc. of workshop on semantic evaluation*, 2010, pp. 190–193.

[30] K. W. Boyack and R. Klavans, "Co-citation analysis, bibliographic coupling, and direct citation: Which citation approach represents the research front most accurately?" *JASIST*, vol. 61, no. 12, pp. 2389–2404, 2010.

[31] X. Shi, J. Leskovec, and D. A. McFarland, "Citing for high impact," in *Proc. of JCDL*, 2010, pp. 49–58.