Query Independent Dynamic Scholarly Article Ranking

Shuai Ma, Chen Gong, Renjun Hu, Dongsheng Luo, Chunming Hu, Jinpeng Huai
SKLSDE Lab, Beihang University, Beijing, China
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing, China
{mashuai, gongchen, hurenjun, lds1995, hucm, huaijp}@buaa.edu.cn

***Abstract*—Ranking query independent scholarly articles is a critical and challenging task, due to the heterogeneous, evolving and dynamic nature of entities involved in scholarly articles. To do this, we first propose a scholarly article ranking model by assembling the importance of involved entities (*e.g.,* articles, venues and authors) such that the importance is a combination of *prestige* and *popularity* to capture the evolving nature of entities. To compute the prestige of articles and venues, we propose a novel Time-Weighted PageRank that extends traditional PageRank with a time decaying factor based on citation statistics (instead of simple exponential decay). We then develop a batch algorithm for scholarly article ranking, in which we propose a block-wise method for Time-Weighted PageRank in terms of an analysis of the citation characteristics of scholarly articles. We further develop an incremental algorithm for dynamic scholarly article ranking, which partitions graphs into *affected and unaffected areas*, and employs different updating strategies for nodes in different areas. Using real-life data, we finally conduct an extensive experimental study, and show that our approach is both effective and efficient for ranking scholarly articles.**

I. Introduction

Query independent ranking of scholarly articles is an important problem for both research and practice. Generally speaking, a ranking is *a function that assigns each item a numerical score*. Query independent ranking aims to give a static ranking based on the scholarly data only, and is independent of how well articles match a specific query. Such a ranking plays a key role in literature recommendation systems, especially in the *cold start* scenario.

Due to its importance, the problem has drawn significant attentions from both academia [1]–[8] and industry [9]–[11]. In the academic community, the most popular approaches have witnessed a shift from citation-count analysis [1], [12] to graph analysis [2], [3], [6], [7]. Graph-based methods leverage the global structure of bibliographic networks and the interactions between heterogeneous entities, and, hence, are usually better than those based on citation-counts only. Efforts have also been made from the industry. For instance, Google Scholar [10] aims to rank articles in the way researchers do, weighing the full text, where they were published, who they were written by, as well as how often and how recently they have been cited. Besides, Semantic Scholar [9] proposes to use the citation velocity, which is a weighted average number of articles' citations in the last three years.

Scholarly articles are involved with multiple entities such as authors, venues, dates and references. Following the graph-based formalization, scholarly article ranking is essentially a problem of assessing the importance of nodes in a heterogeneous network. However, effective and efficient ranking of nodes in such a large complex network is a challenging task due to the heterogeneous, evolving and dynamic natures of involved entities [13], [14].

First, even if we are only to rank one type of entities (*i.e.,* scholarly articles), the other types of entities (*e.g.,* venues and authors) are closely involved, and, moreover, different types of entities may have different impacts on the ranking of scholarly articles. Second, the importance of articles evolve with time in a complex way [15], [16]. Newly published articles are very likely to have increasing impacts in the next few years, and those published many years ago tend to have decreasing impacts, which conforms to the universal citation pattern of articles such that the number of citations generally grows in the first two to three years, and then declines over the rest of time [16]. Moreover, instead of the universal one, individual articles indeed follow a remarkably diverse set of patterns featured by the time when the number of citations reaches the peaks [16]. Finally, academic data is dynamic and continuously growing. Indeed, the number of articles in MAG has exceeded 126 million, and keeps increasing at around 5.7 million per year [11]. This may possibly cause certain long-term biases into data, *e.g.,* the number of citations increases significantly over time [17], which should be properly considered for scholarly article ranking.

Query independent ranking of scholarly articles is challenging [18], although there exists quite a bit of work on scholarly article ranking, *e.g.,* [1]–[7]. Most previous work exploits the time-dependent information of scholarly data in the form of exponential decay [19]–[22], which fails to fit to the diverse citation patterns of individual articles [16]. Further, to our knowledge, little concern has been paid to dynamic scholarly article ranking except [23] with a strong and impractical assumption that there are no citations between papers published in the same years.

Contributions & Roadmap. To this end, we propose an effective and efficient approach for query independent scholarly article ranking in a dynamic environment.

(1) We first propose a Scholarly Article Ranking model, referred to as SARank, by assembling the importance of three classes of entities (articles, venues and authors) for scholarly

article ranking (Section II). The importance is a combination of *prestige* and *popularity* to capture the evolving nature of entities. To compute the prestige of articles and venues, we propose a novel *Time-Weighted PageRank* with a time decaying factor based on citation statistics (instead of simple exponential decay), and the prestige of authors is the average prestige of all their published articles. The popularity of an article is the sum of all its citations' freshness (how close to the current year), while the one of venues and authors is the average popularity of their associated articles. To the best of our knowledge, the Time-Weighted PageRank is among the first to incorporate the diverse citation patterns of individual articles for scholarly article ranking.

(2) We then develop a batch algorithm for scholarly article ranking (Section III), in which we propose a block-wise method for Time-Weighted PageRank in terms of an analysis of the citation characteristics of scholarly articles. We further develop an incremental algorithm for dynamic scholarly article ranking (Section IV), which partitions graphs into *affected and unaffected areas*, and employs different updating strategies for nodes in affected and unaffected areas.

(3) Using three real-life scholarly datasets (AAN, DBLP and MAG) and two sets of ground-truth (RECOM and PFCTN), we finally conduct an extensive experimental study (Section V). (a) We find that, with RECOM and PFCTN, our model SARank improves the pairwise accuracy [24] over (PRank [25], FRank [21], HRank [5]) by (13.5%, 6.8%, 4.8%) and (12.0%, 3.0%, 3.2%) on AAN, (12.7%, 5.0%, 4.9%) and (14.0%, 6.5%, 4.6%) on DBLP, and (6.5%, 2.5%, 2.2%) and (13.4%, 6.0%, 2.4%) on MAG, on average, respectively. (b) Our batch algorithm batSARank and incremental algorithm incSARank are also efficient. Indeed, incSARank is on average (1.7, 2.8, 116) and (2.0, 4.4, 245) times faster than (batSARank, FRank, HRank) on the large DBLP and MAG, respectively.

II. RANKING MODEL

In this section, we first present Time-Weighted PageRank for evaluating the importance of entities, defined as a combination of the prestige and popularity, and then introduce our ranking model SARank that assembles the importance of articles, venues and authors involved in scholarly articles.

A. Time-Weighted PageRank

We first present Time-Weighted PageRank (TWPageRank) based on citation statistics, as the direct use of PageRank for ranking scholarly articles is problematic as discussed below.

(1) Different articles typically have different impacts in practice, on which there is a need to differentiate, while PageRank essentially assumes equal impacts.

(2) The semantics of citation relationships are time-dependent, which means that citations at different periods of time potentially reveal different information. This has already been exploited for scholarly article ranking [19], [20], [22], while PageRank does not consider temporal information at all.

Time-Weighted PageRank (TWPageRank). Most previous work simply exploits temporal information in the form of exponential decay [19]–[22]. We rethink the usage of time information in terms of the impacts of scholarly articles. Recall that [16] categorizes all articles into six citation patterns featured by the time when the articles reach their citation peaks. These patterns are *PeakInit* with the citation-count peak in the first five years (but not the first year) after publication, *PeakMul* with distinct multiple peaks, *PeakLate* with a single peak in at least five years after publication, *MonDec* with monotonically decreasing citations, *MonIncr* with monotonically increasing citations, and, finally, *Other* for articles whose average numbers of citations per year are less than 1. Taking the number of citation as an indicator of the impacts of articles [1], [15], the impacts are time-dependent, but not simply in the form of exponential decay. In general, *the impact of an article tends to decay with time after the peak time only*. That is, the impacts of articles directly decay with time only for ones in *MonDec*, and decay with time after the peak time for ones in *PeakInit*, *PeakLate* and *PeakMul*, and do not decay for ones in *MonIncr*. Note that *each individual article has its own peak time* as articles may reach their citation peaks in different patterns and time.

Based on the above discussion, we propose TWPageRank that evaluates the prestige of nodes (*e.g.,* scholarly articles) in a directed graph, such that each node is attached with time information. It differs from PageRank by weighting the influence propagation using the *impact weights on edges*, which represent the relative amounts of time-dependent prestige that should be propagated from the edge sources to targets. Formally, the impact weight on a directed edge (u, v), *i.e.,* an edge from u to v, is defined as:

$$w(u,v) = \begin{cases} 1 & T_u < Peak_v \\ e^{\sigma(T_u - Peak_v)} & T_u \geq Peak_v, \end{cases} \quad (1)$$

where T_u is the time of node u, $Peak_v$ is the peak time of node v after which the impact weights of edges to v decay with time, and σ is a negative number controlling the decaying speed of the impacts. By default, Eq. (1) uses years as its time granularity. Note that the time decaying factor σ is introduced to provide flexibility for TWPageRank in various applications, and its value is typically within a small interval, *e.g.,* $[-2, 0]$, such that $w(u, v)$ does not decay when $\sigma = 0$ and already decays more than a half per year when $\sigma = -1$. For the sake of completeness, we further set $w(u, v)$ to 0 if there does not exist an edge from u to v in the graph.

For scholarly article ranking, T_u is the publication time of article u and $Peak_v$ should be ideally set to the time when article v has the highest impacts. Basically, it could be the year when article v obtains the largest number of citations. However, this will let $Peak_v$ bias to recent years. Recent work observes that the volume of scientific publications as well as the number of references grow exponentially with time [17], [26]. We also collect the references statistics on three scholarly datasets, shown in Fig. 1, where an exponential distribution can be observed before 2012, and the number decreases then due to data incompleteness. Hence, we compute the scaled number of citations $\Psi_v^{(t)} = \Phi_v^{(t)} / \log Z^{(t)}$ where $\Phi_v^{(t)}$ and $Z^{(t)}$

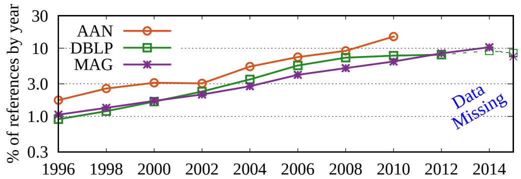


Figure 1. Reference statistics of scholarly articles

are the number of citations of article v at year t and the total number of references at year t, respectively, and set $Peak_v$ to the year that maximizes $\Psi_v^{(t)}$.

The update rule of TWPageRank is:

$$PR(v) = d \sum_{(u,v)\in E} \frac{w(u,v)PR(u)}{W(u)} + \frac{1-d}{n}, \quad (2)$$

where $PR(u)$ and $PR(v)$ are the TWPageRank scores of u and v, respectively, E is the set of edges, $W(u) = \Sigma_v w(u,v)$ is the sum of the impact weights on all edges from u, n is the number of nodes and d is a damping parameter in $(0,1)$. By Eq. (2), we can see that prestige is based on the impact weights, not equally distributed.

Correspondingly, the matrix form of the update rule is:

$$PR^{(t)} = dM^T PR^{(t-1)} + (1-d)e/n. \quad (3)$$

Here $PR^{(k)}$ is the TWPageRank vectors after k iterations, M is the transition matrix such that $M_{u,v} = w(u,v)/W(u)$ and e is an n-dimensional all-one vector $[1]_{n\times 1}$.

The linear system in Eq. (3) is equivalent to *irreducible* and *aperiodic* Markov chains [27], which guarantees the following.

Proposition 1: *TWPageRank converges to a unique vector on any graph, regardless of the initial vector.* □

B. Ranking with Importance Assembling

In our model, the importance is defined as a combination of the prestige and popularity. Intuitively, prestige favors those with many citations soon after the publication of articles or associated articles of venues and authors, and popularity favors those with recent citations. Both prestige and popularity capture the temporal nature of entities.

Our ranking model SARank, illustrated in Fig. 2, assembles the importance of article, venue and author entities for scholarly article ranking, which is computed by the citation, venue and author components, respectively. We next introduce the details of the three components.

Citation component. The first component computes the importance of articles using the citation information.

A *citation graph* $G^c(V^c, E^c)$ is firstly constructed using the citation information such that (a) a node in V^c denotes an article, (b) a directed edge (u,v) in E^c denotes that u cites v, and (c) each node is associated with two types of time information: the publication year and the latest year having the largest number of citations.

(1) The prestige of articles is derived by applying TWPageRank on the citation graph G^c, and each article v is assigned the corresponding TWPageRank score as its prestige $Prs_c(v)$.

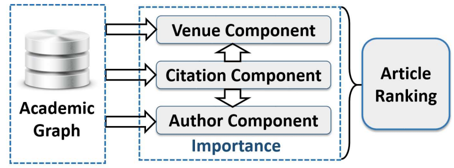


Figure 2. Ranking model SARank

(2) The popularity of an article is the sum of all its citations' freshness, *i.e.,* the closeness to the current year:

$$Pop_c(v) = \sum_{(u,v)\in E^c} e^{\sigma(T_0 - T_u)}. \quad (4)$$

Here T_0 is the current year, *i.e.,* the latest T_u among all articles in V^c, σ is the negative decaying factor used in Eq. (1), and $e^{\sigma(T_0-T_u)}$ represents the freshness of citation (u,v).

Intuitively, the more recent citations an article has, the higher its popularity is, no matter how long it has been published. Different from $Peak_v$ in Eq. (1), the bias of references has little impacts on the popularity. Moreover, the motivation of popularity is to assign higher ranks to articles of recent attention, which is somehow biased to recent articles. Hence, we do not explicitly handle the bias for popularity computation. Note that the popularity is also normalized such that the sum of all articles is equal to 1, similar to the prestige produced by TWPageRank.

(3) The prestige and popularity are finally combined to produce the importance of articles. Intuitively, an important article is both prestigious and popular. Hence, the *citation importance score* $Imp_c(v)$ of an article is defined as a weighted combination of its prestige and popularity:

$$Imp_c(v) = Prs_c(v)^{\lambda} Pop_c(v)^{1-\lambda}, \quad (5)$$

where $\lambda \in [0,1]$ is the importance weighting factor. The rationales behind Eq. (5) are as follows. (a) Prestigious articles with many recent citations are ranked at the top, as researchers are very willing to find them; (b) Prestigious articles with rare current citations are ranked lower, as researchers may lose interests in these old articles; And (c) articles with many recent citations are ranked higher, as researchers have potential interests in those of recent attention.

Venue component. The second component computes the importance of venues with their associated articles. As the importance of a venue evolves with time, we treat the venue in each year individually, and its importance is the sum of importance in all individual years.

A *venue graph* $G^v(V^v, E^v)$ is firstly constructed using the citation information among venues such that (a) a node in V^v represents a venue in a specific year, (b) a direct edge (s,t) in E^v denotes that there exist articles published in venue (in a specific year) s citing articles published in venue (in a specific year) t, and (c) we use the *impact weight* to denote the weight $w_v(s,t)$ from venues s to t, which is the sum of the impact weights from articles published in s to t, *i.e.,*

$$w_v(s,t) = \sum_{u\in C(s), v\in C(t)} w(u,v). \quad (6)$$

Here, $C(s)$ and $C(t)$ are the sets of articles published in s and t, respectively, and $w(u, v)$ is the impact weight of edge (u, v) produced in the citation component.

The prestige of a venue in a specific year is computed using the impact weights and the update rule in Eq. (2), and the popularity of a venue in a specific year is defined as the average popularity of its articles. The prestige and popularity are combined to derive the importance of a venue in a specific year in the same way as the citation component. Finally, the importance of a venue is treated as the *venue importance score* for all articles published in this venue.

Author component. The author component computes the importance of authors with their published articles. Similar to the venue component, we evaluate the importance of each author, and compute the average importance of the authors of an article as its *author importance score*.

However, the resulting author citation graph to compute the prestige is typically too large to handle. Hence, we evaluate the prestige of an author, by using the average prestige of all articles published by the author. Similar to the venue component, the popularity of an author is also defined as the average popularity of her/his published articles. Finally, the prestige and popularity are combined to derive the importance in the same way as the citation component.

Ranking with importance assembling. The aforementioned importance is finally assembled to produce the final ranking, as illustrated in Fig. 2. Before assembling, each component is properly scaled such that the average citation importance score, venue importance score and author importance score are the same. Let the scaled importance scores of article v be $R_c(v)$, $R_v(v)$, and $R_a(v)$ from the citation, venue and author components, respectively. The final ranking score $R(v)$ of an article v is aggregated as follows:

$$R(v) = \alpha R_c(v) + \beta R_v(v) + (1 - \alpha - \beta) R_a(v). \tag{7}$$

Here aggregating parameters α and β and value $(1 - \alpha - \beta)$ regularize the contributions of the citation, venue and author information. Intuitively, these parameters indicate the intensity of the correlation between the importance of scholarly articles and the specific information.

Remarks. It is also possible to train a discriminative model which directly optimizes certain loss functions for ranking, *e.g.,* [24] for Web pages. Alternatively, this work follows the graph-based formalization, and further develops efficient batch and incremental algorithms based on graphs for scholarly article ranking (Sections III & IV).

III. Ranking Computation

In this section, we present our batch algorithm for computing scholarly article ranking based on our model SARank.

A. Algorithm Framework

Our batch algorithm batSARank combines the importance scores computed by the citation, venue and author components with Eq. (7). It takes as input academic graph data D and an iteration threshold ϵ and returns the scholarly article ranking of D. It first constructs the citation and venue graphs $G^c(V^c, E^c)$ and $G^v(V^v, E^v)$. Then it computes the prestige and popularity of citation, venue and author components. Finally, it combines the prestige and popularity of the three components to produce the final ranking with Eq. (7).

For popularity computation, it is easy to see that (a) the popularity of articles can be computed by scanning through all citations once and adding the freshness of citations to their corresponding articles, by Eq. (4), and (b) the popularity of venues in a specific year or authors is computed by averaging the popularity of the articles published in the venues or by the authors. That is, the popularity computation can be done by scanning through all citations once.

For prestige computation, as the one of authors is defined as the average prestige of their published articles, it suffices to scan through all author-article relationships for computing the prestige of authors. The prestige of articles and venues in a specific year is computed by TWPageRank on citation graphs and venue graphs, which is usually computed in an iterative manner [25] and is the most expensive computation. Hence, the key of the computation of our approach is a good solution for computing TWPageRank.

B. TWPageRank Computation

The main result here is to speed up computing TWPageRank by exploiting the *temporal order* of scholarly articles.

Claim 2: *A block-wise PageRank computation method [28] is a good choice for TWPageRank on scholarly data.* □

The main idea of the block-wise PageRank computation is that each strongly connected component (SCC) of the input graph is treated as a block, and blocks are processed one by one following the *topological order* of the block-wise graph, *i.e.,* each node represents a block of the original graph [28]. We next show Claim 2 by introducing and analyzing such a block-wise computation method.

Block-wise algorithm batTWPR. It takes as input a citation or venue graph G and an iteration threshold ϵ, and returns the TWPageRank vector of G. Moreover, to process an SCC instead of the entire graph, the edges of the citation or venue graph $G(V, E)$ are partitioned into the sets E_i and E_a of edges inside and across SCCs such that $E_i \cap E_a = \emptyset$ and $E = E_i \cup E_a$. The update rule in Eq. (3) is revised accordingly to treat E_i and E_a separately as follows.

$$PR(v) = d \sum_{(u,v) \in E_i} M_{u,v} PR(u) + d \sum_{(u,v) \in E_a} M_{u,v} PR(u) + \frac{1-d}{n}. \tag{8}$$

It first computes the block-wise graph G' by treating SCCs in G as single nodes, and then derives a topological order O of nodes in G'. It then processes each SCC in the topological order O with Eq. (8). Finally, it returns the TWPageRank vector. When processing an SCC, it iteratively updates the TWPageRank scores of the nodes in the SCC, and the iteration continues until the sum of TWPageRank score changes is less than $\epsilon \cdot \frac{|scc|}{|V|}$, where $|scc|$ is the number of nodes in the SCC.

Table I
STATISTICS OF CITATION/VENUE GRAPHS AND WEB GRAPHS [30]

Graphs	Nodes	Edges	Largest $\lvert SCC\rvert$	SCC edge ratio
AAN	18,041/565	82.9K/22.5K	20/18	0.9%/2.8%
DBLP	3.14M/56K	14.3M/7.1M	23/1.5	1.6%/2.1%
MAG	127M/584K	526M/162M	351/10K	0.1%/1.8%
web-BS	685,230	7,600,595	334,857	59.51%
web-G	875,713	5,105,039	434,818	66.98%

Note that there must exist a topological order O, as the block-wise graph is a directed acyclic graph [29].

Corollary 3: *The vector PR returned by* batTWPR *converges such that $||PR - PR^*||_1 < \epsilon$ where vector PR^* is the convergent TWPageRank vector [28].* □

Proof Sketch: We first prove that the sum of changes after another iteration from PR, *i.e.,* $||dM^T PR + (1-d)e/n - PR||_1$, is smaller than ϵ, and then prove that $||PR^* - PR||_1$ is smaller than the sum of changes. □

Analysis of the block-wise algorithm. While similar block-wise algorithms were originally proposed for Web graphs [28], we next show that they are even better for the TWPageRank computation associated with scholarly data. The block-wise graph and its topological order can be done in $O(|V| + |E|)$ time [29], and updating TWPageRank scores takes $O(|V| + |E_a| + t|E_i|)$ time as the edges in E_a are only scanned once. From these, we have the following.

Lemma 4: *Given a citation graph or venue graph $G(V, E)$, algorithm* batTWPR *runs in $O(|V|+|E_a|+t|E_i|)$ time, where t is the maximum number of iterations among all* SCCs. □

Recall that t is very likely to be in the scale of tens to hundreds [25]. Hence, the efficiency of block-wise algorithm batTWPR is mainly affected by $|E_i|$, *i.e.,* the smaller $|E_i|$ is, the faster algorithm batTWPR is.

It is well-known that article citations obey a natural temporal order, *i.e.,* an article only cites those published earlier, and it is really rare for the mutual citations between two articles published in the same time. That is, $|E_i|$ is essentially small for the citation graphs of scholarly articles. We also collect the statistics of citation graphs, venue graphs and Web graphs, shown in Table I, to verify our observation, where Web graphs are extracted from berkely.edu and stanford.edu domains in 2002 and from the Google programming contest in 2002, respectively [30]. Due to the existence of the "bow tie" structure and the giant SCC in Web graphs [31], the edge ratios $|E_i|/|E|$ are greater than 59% for both Web graphs. In contrast, the SCCs in citation and venue graphs are quite small as a result of the temporal order, and $|E_i|/|E|$ is less than 3% for all tested citation and venue graphs. This specific structure in scholarly data has long been ignored in the literature, which has a large impact on the running time. Taking $t = 100$ for example, algorithm batTWPR needs to scan through $3|E|$ and $4|E|$ edges on citation and venue graphs, respectively, while over $59|E|$ edges on Web graphs.

By Corollary 3, Lemma 4 and the above analysis of our block-wise algorithm, we have informally established Claim 2.

Time & space complexity analyses of the batch algorithm. By Lemma 4 and the analyses in Section III-A, one can verify that algorithm batSARank takes $O(|V^c|+|V^v|+|E^c_a|+|E^v_a|+t|E^c_i|+t|E^v_i|+|PA|)$ time, where $|PA|$ is the total number of author-article relationships, and takes $O(7|V^c|+2|V^{c\prime}|+4|V^v|+2|V^{v\prime}|+|E^c|+|E^{c\prime}|+|E^v|+|E^{v\prime}|+2|A|+|D|)$ space, where $|A|$ is the number of authors and $|D|$ is the size of academic graph data D.

The key of algorithm batSARank is to compute TWPageRank with algorithm batTWPR. Compared with the traditional power method [25], our block-wise algorithm batTWPR uses $(2|V^{c\prime}|+|E^{c\prime}|)$ and $(2|V^{v\prime}|+|E^{v\prime}|)$ extra space to store the block-wise graphs $G^{c\prime}(V^{c\prime}, E^{c\prime})$ of G^c and $G^{v\prime}(V^{v\prime}, E^{v\prime})$ of G^v and their topological orders, while speeds up computation by $O((t-1)(|E^c_a|+|E^v_a|))$.

IV. DYNAMIC RANKING COMPUTATION

Scholarly articles are dynamic and continuously growing, and it is impractical to recompute ranking from scratch once they get updated. In this section, we present an incremental algorithm for our ranking model SARank.

A. Incremental Algorithm Framework

Our incremental algorithm incSARank incrementally computes the popularity and prestige of scholarly articles. We consider that an update $\Delta = \Delta V \cup \Delta E$ is added to a (citation or venue) graph $G(V, E)$, and the resulting graph is $G^+(V \cup \Delta V, E \cup \Delta E)$, where ΔV is a set of nodes with $\Delta V \cap V = \emptyset$, and ΔE is a set of directed edges on ΔV and from ΔV to V only, as article citation relationships obey a natural temporal order, *i.e.,* an article only cites those published earlier, and it is rare for the mutual citations between two articles published in the same time.

Incremental popularity computation. The popularity of venues and authors is computed along the same lines as their batch counterparts of algorithm batSARank, as almost all venues and authors are affected by the definitions of the popularity of venues and authors.

As the popularity of articles is defined as the freshness sum of all their citations, it is convenient to maintain in a dynamic scenario. Consider an updated citation graph $G^{c,+}(V^c \cup \Delta V^c, E^c \cup \Delta E^c)$ of $G^c(V^c, E^c)$, and the updated popularity $Pop^+_c(v)$ can be computed as:

$$Pop^+_c(v) = Pop_c(v)e^{\sigma(T^+_0 - T_0)} + \sum_{(u,v)\in\Delta E^c} e^{\sigma(T^+_0 - T_u)}, \quad (9)$$

where $Pop_c(v)$ (resp. $Pop^+_c(v)$) is the popularity of node v on G^c (resp. $G^{c,+}$), and T_0 (resp. T^+_0) is the current time in G^c (resp. $G^{c,+}$). By Eq. (9), it is easy to see that it takes $O(|V^c| + |\Delta V^c| + |\Delta E^c|)$ time to update the popularity.

Incremental prestige computation. The prestige of authors is computed along the same lines as the batch algorithm batSARank, as almost all authors are affected by the definition of the prestige of authors. For articles and venues, we propose an incremental algorithm to maintain their prestige.

Input: An update $\Delta = \Delta V \cup \Delta E$, TWPageRank vector PR of G, and the topological order O of the block-wise graph G'.
Output: TWPageRank vector PR^+ of the updated graph G^+.
1. G'_C := the block-wise graph of G_C;
2. ΔO := topological order of G'_C; $O^+ := \Delta O/O$;
3. label SCCs of G_C as C and SCCs of G with outgoing edges having weight changes as B, the remaining SCCs of G as A
4. **for** each node v' following O^+ **do**
5. $\quad$ scc := the corresponding SCC of v';
6. $\quad$ **if** scc is labeled as C **then**
7. $\quad\quad$ update $PR^+(v)$ ($v \in scc$) following algorithm batTWPR;
8. $\quad\quad$ label SCC w' as B with $w' \in G'$ and $(v', w') \in E^{+\prime}$;
9. $\quad$ **else if** scc is labeled as B **then**
10. $\quad\quad$ update $PR^+(v)$ where $v \in scc$ with Eq. (10) until the sum of TWPageRank score changes is less than $\epsilon \cdot \frac{|scc|}{|V^+|}$;
11. $\quad\quad$ label SCC w' as B with $(v', w') \in E'$;
12. $\quad$ **else** $PR^+(v)$:=$PR(v) \cdot n/n^+$ where $v \in scc$;
13. **return** PR^+.

Figure 3. Algorithm incTWPR for incremental TWPageRank

B. Incremental TWPageRank Computation

Consider a citation or venue graph $G(V, E)$, its TWPageRank vector PR and the topological order O of its block-wise graph. Given an update $\Delta = \Delta V \cup \Delta E$ to G, the incremental prestige computation for articles and venues in a specific year is to compute the TWPageRank vector PR^+ on the updated graph $G^+(V \cup \Delta V, E \cup \Delta E)$.

Auxiliary data structure maintenance. Two auxiliary data structures in the batch algorithm batTWPR need to be maintained : (a) on the block-wise graph, a mapping that, given a node of graph G, returns the index of the SCC to which it belongs, and (b) the topological order of the nodes in the block-wise graph. Observe that these auxiliary data structures can be easily maintained as follows.

(1) The block-wise graph of G^+ needs to be computed, whose SCCs consist of the SCCs in G and SCCs in the induced subgraph $G^+[\Delta V]$, as the edges of ΔE are on nodes in ΔV and from ΔV to V only. Hence, only those new SCCs in $G^+[\Delta V]$ need to be computed.

(2) The updated topological order $O^+ = \Delta O/O$, where ΔO is the topological order of the block-wise graph of induced subgraph $G^+[\Delta V]$. Hence, only ΔO needs to be computed. One can easily verify the following.

Proposition 5: *$O^+ = \Delta O/O$ is indeed a valid topological order of the block-wise graph of G^+.* □

Proof Sketch: We prove that for each edge (u, v) in the block-wise graph of G^+, node u comes before v in O^+, and the conclusion follows by the definition of topological order. □

Analyses of affected and unaffected areas. The TWPageRank vector PR of graph G is mainly affected in two ways.

(1) Let $V_{B,1} \subseteq V$ be the set of nodes reachable from the newly added nodes ΔV, $V_{B,2} \subseteq V$ be the set of nodes with outgoing edges having weight changes, and $V_{B,3} \subseteq V$ be the set of nodes reachable from $V_{B,2}$. Then $V_B = V_{B,1} \cup V_{B,2} \cup V_{B,3}$ is obviously the set of nodes in G affected by the update Δ. TWPageRank scores on V_B are re-iterated as follows, where

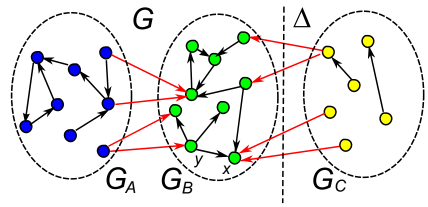


Figure 4. An example of incremental TWPageRank computation

notations with superscript '+' are defined on G^+.

$$\begin{aligned} PR^+(v) &= d \sum_{(u,v)\in E_i^+} M^+_{u,v} PR^+(u) + d \sum_{(u,v)\in E_a^+} M^+_{u,v} PR^+(u) \\ &+ \frac{n}{n^+}\Big(PR(v) - d \sum_{(u,v)\in E_i} M_{u,v} PR(u) - d \sum_{(u,v)\in E_a} M_{u,v} PR(u)\Big). \end{aligned} \tag{10}$$

(2) Let $V_A = V \backslash V_B$. Since nodes in V_A are not reachable from newly added or affected nodes, V_A is essentially not affected by the update Δ. And TWPageRank scores on V_A only need to scale with constant n/n^+.

Let $G_A = (V_A, E_A)$, $G_B = (V_B, E_B)$ and $G_C = (V_C, E_C)$, respectively, and let E_{AB} and E_{CB} be the sets of edges from G_A to G_B and from G_C to G_B, respectively. In this way, graph G^+ is divided into subgraphs $\{G_A, G_B, G_C\}$ and edge sets $\{E_{AB}, E_{CB}\}$. We then have $G_C = G^+[\Delta V]$, $\Delta E = E_C \cup E_{CB}$, $V = V_A \cup V_B$ and $E = E_A \cup E_B \cup E_{AB}$.

Incremental algorithm incTWPR. We now present our incremental algorithm for TWPageRank, shown in Fig. 3.

It takes as input an update Δ and the previous results on the original graph $G(V, E)$, and returns the TWPageRank vector of the updated graph G^+. It first incrementally computes the topological order O^+ (lines 1–2). After that, it labels the newly added SCCs with C and existing SCCs with A or B, depending on whether the existing SCCs have weight changes on outgoing edges (line 3). It then processes each SCC in the order O^+ such that the TWPageRank scores of nodes in each SCC are updated according to the labels (lines 4–12), and finally returns the TWPageRank vector (line 13).

When processing V_B with Eq. (10), *edges in E_{AB} can be skipped* since $PR^+(u) = n/n^+ \cdot PR(u)$ for $u \in V_A$ and $M_{u,v} = M^+_{u,v}$ for $(u, v) \in E_{AB}$. Besides, we use $n/n^+ \cdot PR$ as the initial vector. Both of them can speed up the computation.

Example 1: Figure 4 illustrates an example of incremental TWPageRank computation. Consider an update Δ on the original graph G. It is obvious that the update Δ has no impacts on the SCCs of G, and O^+ defined earlier is a valid topological order of $G^{+\prime}$. The original graph G is then partitioned into affected and unaffected areas, and subgraphs G_A, G_B and G_C are associated with node sets V_A, V_B and ΔV, respectively. Here edge weight on (y, x) changes due to the change of the citation peak time of node x, and, hence, node y as well as all nodes reachable from y are included in G_B. When updating the TWPageRank scores, following O^+, scores of nodes in G_C, G_B and G_A are computed by iterations from scratch, by iterations with Eq. (10) using the existing TWPageRank vector and by scaling, respectively. □

Theorem 6: *The TWPageRank vector PR^+ returned by* incTWPR *converges such that $||PR^+ - PR^*||_1 < \epsilon$, where*

Table II

STATISTICS OF AFFECTED/UNAFFECTED AREAS GIVEN A YEARLY UPDATE, *i.e.,* ARTICLES OF 2011 ON AAN AND 2015 ON DBLP AND MAG, RESPECTIVELY.

Statis.	Citation graphs on			Venue graphs on		
	AAN	DBLP	MAG	AAN	DBLP	MAG
$\|V_A\|$	47.4%	52.3%	69.2%	2.1%	8.7%	12.4%
$\|V_B\|$	46.8%	40.0%	26.3%	92.0%	84.8%	84.6%
$\|V_C\|$	5.8%	7.8%	4.5%	5.8%	6.4%	3.0%
$\|E_A\|$	3.0%	2.4%	0.9%	0.0%	0.0%	0.0%
$\|E_{AB}\|$	26.5%	30.2%	26.6%	1.2%	0.2%	0.1%
$\|E_B\|$	59.8%	59.3%	65.5%	88.6%	92.3%	92.6%
$\|E_{CB}\|$	10.4%	7.2%	7.0%	10.0%	7.3%	7.1%
$\|E_C\|$	0.3%	0.9%	0.1%	0.2%	0.2%	0.1%

PR^* is the convergent TWPageRank vector. □

Proof Sketch: Assume a topological order $v'_1/\ldots/v'_l$ of the block-wise graph $G^{+\prime}$ where $l = |O^+|$. It suffices to prove by induction that the sum of changes of $PR^+(v)$ ($v \in scc_k$) is no more than $\epsilon|scc_k|/|V^+|$ for scc_k of v'_k ($k \in [1, l]$). □

Observe that (a) a topological order of G'_C can be computed in $O(|V_C|+|E_C|+|E_{CB}|)$ time, (b) updating the TWPageRank scores of nodes in subgraphs G_B and G_C costs $O(|V_B \cup V_C| + |E_{B,a} \cup E_{C,a} \cup E_{CB}|) + t^+|E_{B,i} \cup E_{C,i}|)$ time, and , finally, (c) updating the scores of nodes in G_A costs $O(|V_A|)$ time. From these, the following holds.

Proposition 7: *Given an update $\Delta = \Delta V \cup \Delta E$ of citation or venue graph $G(V, E)$, the TWPageRank vector of G and the topological order of G', algorithm* incTWPR *runs in $O(|V \cup \Delta V| + |E_B \cup E_C \cup E_{CB}| + t^+|E_{B,i} \cup E_{C,i}|)$ time.* □

By Propositions 1 & 5 and Theorem 6, one can easily verify the correctness of algorithm incTWPR. Note that a) algorithm incTWPR computes SCCs and derives the topological order based on Δ only, instead of G^+, b) it skips edges in $E_A \cup E_{AB}$ when updating the scores of nodes in G_A and G_B, and c) the number t^+ is very likely smaller than the number t of batTWPR when updating scores of nodes in G_B. All these make incTWPR faster than batTWPR even though they have very similar time complexity.

Time & space complexity analyses of the incremental algorithm. By the analyses above, the time complexity of incSARank is the same as batSARank, except that incSARank saves $O(|E^c_A \cup E^c_{AB}|)$ and $O(|E^v_A \cup E^v_{AB}|)$ time on the updated citation and venue graphs. And its space complexity is also the same as batSARank, except that it uses $(|V^c \cup \Delta V^c| + |V^v \cup \Delta V^v|)$ extra space to store the affected/unaffected areas and $(|E^c \cup \Delta E^c| + |E^v \cup \Delta E^v|)$ extra space to store the original edge weights before update.

Despite of its similar time complexity to batSARank, algorithm incSARank typically achieves a substantial efficiency improvement over batSARank, according to our statistics of affected/unaffected areas shown in Table II. (a) It saves $O(|V^c| + |E^c|)$ and $O(|V^v| + |E^v|)$ time when maintaining SCCs and the topological order based on Δ^c and Δ^v only, where ($|V^c|$, $|E^c|$, $|V^v|$, $|E^v|$) are more than (92%, 89%, 93%, 89%) of ($|V^{c,+}|$, $|E^{c,+}|$, $|V^{v,+}|$, $|E^{v,+}|$) on all tested graphs; (b) It saves $O(|E^c_A \cup E^c_{AB}|)$ time when updating scores on V^c, where edges in $E^c_A \cup E^c_{AB}$ account for more than 28% of total edges; (c) It saves $O(|E^c|)$ time when computing popularity of articles, which accounts for more than 89% of $|E^{c,+}|$; And, finally, (d) It is likely to compute TWPageRank scores on V^c_B and V^v_B with less iterations.

V. EXPERIMENTAL STUDY

In this section, we present an extensive experimental study of our approach SARank, compared with competitive methods. Using three real-life scholarly datasets (AAN, DBLP and MAG) and two sets of ground-truth (RECOM and PFCTN), we conducted four sets of experiments to evaluate: (1) the effectiveness of SARank, (2) the efficiency of our batch algorithm batSARank and incremental algorithm incSARank, and (3) the impacts of parameters.

A. Experimental Settings

We first present the settings of our experimental study.

Datasets. We chose three datasets to test our approach.
(1) AAN records the collection of computational linguistics articles published at ACL conferences from the year of 1965 to 2011 [5]. It contains 18,041 articles, 14,386 authors, 273 venues and 82,944 citations.
(2) DBLP records articles in the computer science domain from 1936 to 2016 [32]. It contains 3.14 million articles, 1.74 million authors, 11,619 venues and 6.38 million citations.
(3) MAG records articles of various disciplines from 1800 to 2016 [11]. It contains around 127 million articles, 115 million authors, 24,024 venues and 529 million citations.

The average number of citations on DBLP is much less than the other two, we hence added part of the missing citations by title matching on MAG, and, finally, the total number increased to 14.26 million. These datasets were further cleaned by deleting self-citations and citations from old articles to new ones, which accounted for (0.1%, 0.8%, 0.4%) of the total citations on (AAN, DBLP, MAG), respectively.

Accuracy metric and ground-truth. We adopted the *pairwise accuracy* introduced by Microsoft [18], [24] to evaluate the ranking quality, *i.e.,* the fraction of times that a ranking agrees with the correct ranking orders of scholarly article pairs. We constructed two sets of ground-truth importance orders of article pairs with RECOM and PFCTN.
(1) RECOM assumes that scholarly articles with more recommendations are of higher importance. We used the numbers of recommendations of 93 articles on AAN [5], and, by exact title matching, generated (2133, 966, 1972) scholarly article pairs on (AAN, DBLP, MAG), respectively.
(2) PFCTN assumes that scholarly articles with more citations are of higher importance. However, the number of all citations is obviously biased to old articles. Some work adopts the number of future citations [19], [20], [33], which is also not appropriate since this gives an estimation of impacts of articles in the near future, not at the query time. Differently, we propose to use the number of past and future citations, where past and future periods span the same length of time such that the number of citations within the two periods reveals

Table III
ACCURACY TESTS WITH RECOM

Datasets	PRank	FRank	HRank	SARank
AAN	0.671	0.738	0.758	**0.805**
DBLP	0.651	0.729	0.730	**0.778**
MAG	0.615	0.655	0.658	**0.680**

the importance of articles at the query time. We hence divided each dataset into two parts by a splitting year where the data before the splitting year (exclusive) were used for ranking and the remaining data as well as the most recent part of ranking data which spans the same time as the remaining data were used to count the numbers of past and future citations. Moreover, articles in the same pair were required to be in similar research fields, by utilizing the Fields-Of-Study information on MAG [11], and published in the same year, similar to [33]. We used all pairs (around 50,000) for AAN, and randomly chose 300,000 pairs for both DBLP and MAG.

Algorithms. We compared our approach with three competitive methods: PRank [25], FRank [21] and HRank [5].
(1) PRank (PageRank) is a classic method that uses only citation information to rank scholarly articles.
(2) FRank (FutureRank) combines citation, temporal and other heterogeneous information to rank scholarly articles.
(3) HRank (HHGBiRank) is a very recent method using both citation and heterogeneous information, such that heterogeneous entities are mutually reinforced based on hypernetworks.

Implementation. All algorithms were implemented with Microsoft Visual C++. For all algorithms, (a) the damping parameter d and the iteration threshold ϵ were fixed to 0.85 and 10^{-8}, respectively, (b) the default splitting years were selected such that the ranking data accounted for around 75% of all, which were 2008 on AAN and 2012 on both DBLP and MAG, and, (c) for the sake of fairness, aggregating parameters of FRank, HRank and SARank were tuned at the granularity of 0.1 and the best results were reported. Moreover, ρ was set to -0.2 for FRank following [21], and the time decaying factor σ and the importance weighting factor λ were set to -1 and 0.5 by default for SARank.

All experiments were conducted on a PC with 2 Intel Xeon E5–2630 2.4GHz CPUs and 64 GB of memory, running 64 bit Windows 7 professional system. The usage of virtual memory was forbidden. When quantity measures are evaluated, the test was repeated over 5 times and the average results are reported.

B. Experimental Results

We next present our findings.

Exp-1: Effectiveness with RECOM. In the first set of our tests, we used ground-truth RECOM to evaluate the effectiveness of our approach. All algorithms used articles published before 2012, since article pairs of RECOM were from this portion of articles. Aggregating parameters were selected as follows: (α, β, γ) = $(0.1, 0.2, 0.2)$ for FRank, (a_{i1}, a_{i2}, a_{i3}) = $(0.6, 0.2, 0.2)$ for HRank ($i \in [1, 3]$), and (α, β) = $(0.1, 0.8)$ for SARank. The results of PairAcc are reported in Table III.

The PairAcc of PRank is much lower than the one of other algorithms, indicating that citation information alone is insufficient for scholarly article ranking, and other information helps to refine the results. Moreover, SARank consistently ranks better than all competitors. Indeed, SARank improves the PairAcc over (PRank, FRank, HRank) by (13.5%, 6.8%, 4.8%) on AAN, (12.7%, 5.0%, 4.9%) on DBLP, and (6.5%, 2.5%, 2.2%) on MAG, respectively.

Exp-2: Effectiveness with PFCTN. In the second set of tests, we used ground-truth PFCTN to evaluate the effectiveness. Aggregating parameters were selected as follows: (α, β, γ) = $(0.7, 0.1,\ 0.2)$ for FRank, (a_{i1}, a_{i2}, a_{i3}) = $(0.3, 0.6, 0.1)$ for HRank ($i \in [1, 3]$), and (α, β) = $(0.8, 0.1)$ for SARank. To evaluate the effectiveness of ranking in different scenarios, we varied three factors in our tests: the splitting year Y_s, the number T_p of published years of articles, and the difference dif of future citation counts. Given Y_s, T_p and dif, we only used article pairs whose articles were published within $[Y_s - T_p, Y_s)$ and the difference of future citation counts was equal to or larger than dif to test the PairAcc.

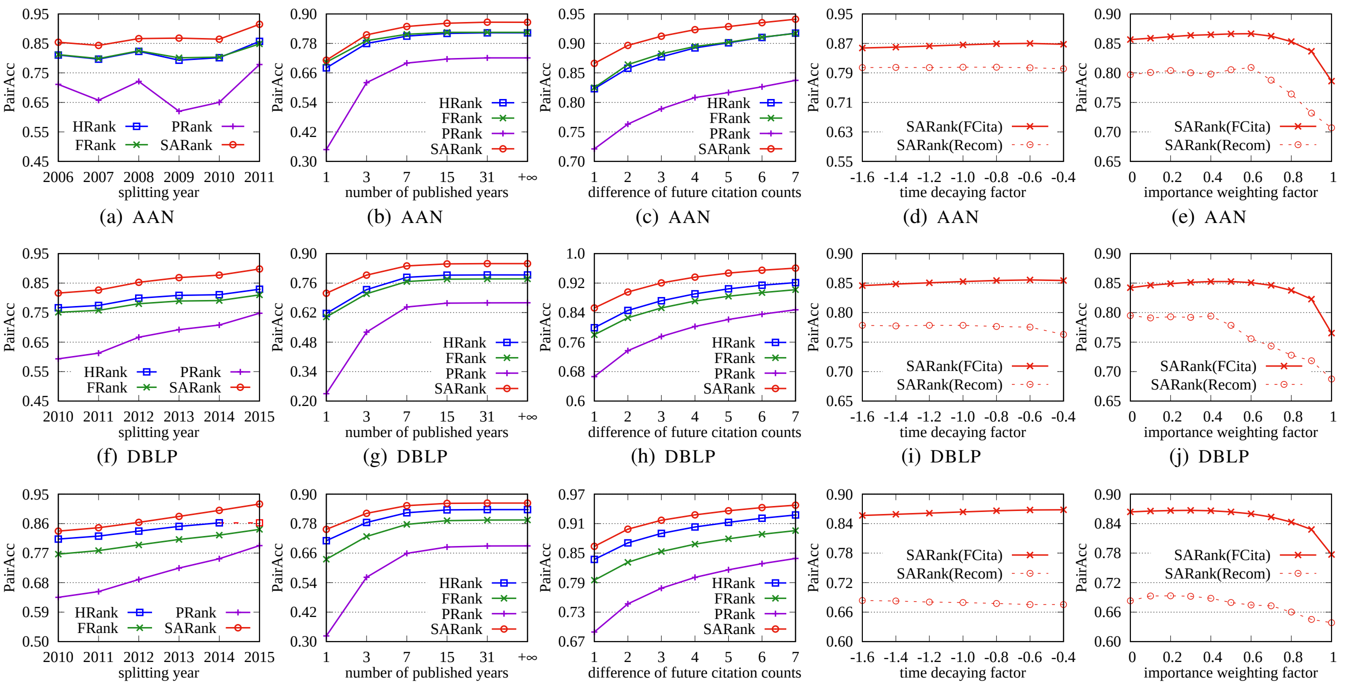
Exp-2.1. To evaluate the effectiveness of ranking *w.r.t. short-term and long-term importance*, we varied the splitting year Y_s from 2006 to 2011 on AAN and from 2010 to 2015 on both DBLP and MAG, while fixed $T_p = +\infty$ and dif = 1, *i.e.,* using all scholarly article pairs. Intuitively, large and small Y_s correspond to short-term and long-term importance, respectively. The results of PairAcc are reported in Figs. 5(a), 5(f) and 5(k), in which the red markers □ in dashed lines mean that HRank ran out of memory.

When varying Y_s, the PairAcc of all algorithms increases with the increment of Y_s on both DBLP and MAG, indicating that it is easier to assess short-term (large Y_s) than long-term (small Y_s) importance. While the results on AAN do not follow this trend, possibly because AAN does not record the complete articles of 2007 and 2009. Moreover, SARank consistently ranks better than all competitors, regardless of assessing short-term or long-term importance. Indeed, SARank improves the PairAcc over (PRank, FRank, HRank) by (17.9%, 5.4%, 5.5%) on AAN, (18.6%, 7.7%, 5.8%) on DBLP, and (16.7%, 7.2%, 2.9%) on MAG, respectively.

Exp-2.2. To evaluate the effectiveness of ranking *w.r.t. the published time of articles*, we varied the number T_p of published years from 1 to $+\infty$, while fixed Y_s to default values of three datasets and $dif = 1$, respectively. The results of PairAcc are reported in Figs. 5(b), 5(g) and 5(l).

When varying T_p, the PairAcc of all algorithms increases with the increment of T_p, since old articles (large T_p) are easier to rank based on adequate information, while new articles (small T_p) are hard to rank with little information available. Moreover, SARank consistently ranks better than all competitors, especially when $T_p \leq 3$, *i.e.,* ranking recently published articles. Indeed, SARank improves the PairAcc over (PRank, FRank, HRank) by (19.0%, 3.1%, 3.9%) on AAN, (25.0%, 8.2%, 6.3%) on DBLP, and (23.6%, 8.3%, 3.2%) on MAG, on average, respectively.

Exp-2.3. To evaluate the effectiveness of ranking *w.r.t. the difference of future citations*, we varied the difference dif of

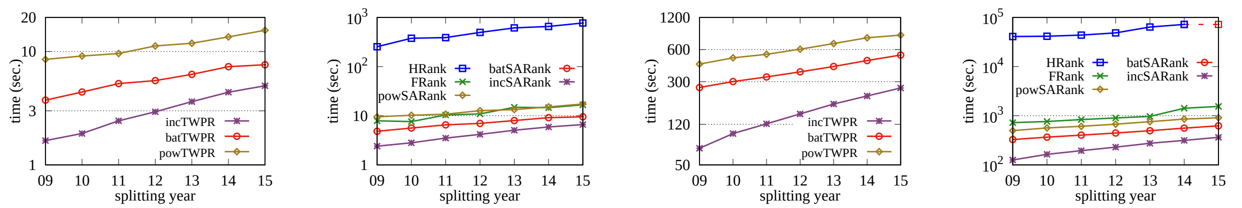


(a) AAN (b) AAN (c) AAN (d) AAN (e) AAN

(f) DBLP (g) DBLP (h) DBLP (i) DBLP (j) DBLP

(k) MAG (l) MAG (m) MAG (n) MAG (o) MAG

Figure 5. Accuracy tests with PFCTN (all) and RECOM ((d)–(e), (i)–(j) and (n)–(o))



(a) TWPageRank (batch vs. inc.) (b) Comparison of ranking algorithms (c) TWPageRank (batch vs. inc.) (d) Comparison of ranking algorithms

Figure 6. Efficiency tests on DBLP ((a)–(b)) and MAG ((c)–(d))

future citation counts from 1 to 7, while fixed Y_s to default values of three datasets and $T_i = +\infty$. The results of PairAcc are reported in Figs. 5(c), 5(h) and 5(m).

When varying dif, the PairAcc of all algorithms increases with the increment of dif, since scholarly article pairs with larger dif are easier to rank. Moreover, SARank consistently ranks better than all competitors, regardless of easy or difficult article pairs. Indeed, SARank improves the PairAcc over (PRank, FRank, HRank) by (12.0%, 3.0%, 3.2%) on AAN, (14.0%, 6.5%, 4.6%) on DBLP, and (13.4%, 6.0%, 2.4%) on MAG, on average, respectively.

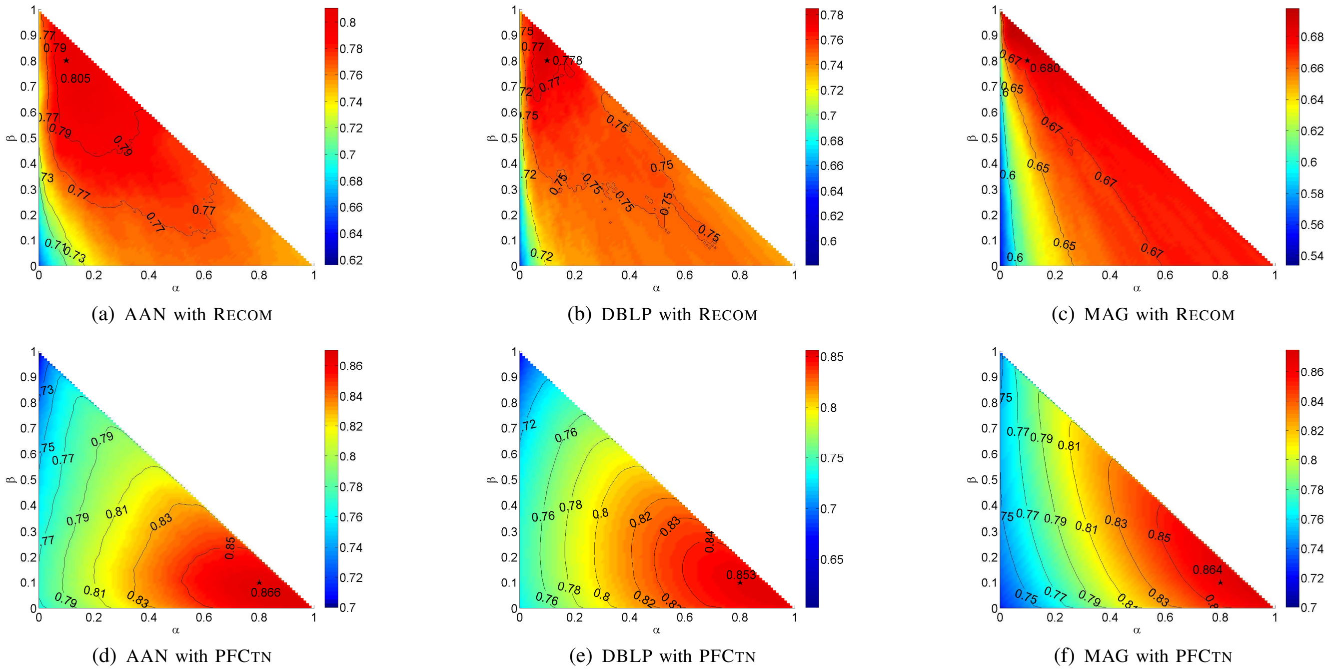
Exp-3: Efficiency. In the third set of tests, we evaluated the efficiency of our algorithms. We compared our algorithms with powTWPR and powSARank, which were the same to batTWPR and batSARank except using power method for TWPageRank computation, and with algorithms FRank and HRank. Here PRank was omitted due to its effectiveness. We varied the splitting year Y_s from 2009 to 2016 and tested the running time on both DBLP and MAG. For incremental algorithms, base and update parts consisted of data before 2008 and within $[2008, Y_s)$, respectively. The results of running time are reported in Fig. 6, where the red markers □ in dashed lines mean that HRank ran out of memory.

When varying Y_s, the running time of all algorithms increases with the increment of Y_s, and our incremental algorithms consistently run faster than all competitors, especially with less update data. For TWPageRank computation, algorithm incTWPR is on average (1.9, 3.8) and (2.5, 4.1) times faster than (batTWPR, powTWPR) on DBLP and MAG, respectively. For scholarly article ranking, algorithm incSARank is on average (1.7, 3.1, 2.8, 117) and (2.0, 3.0, 4.4, 245) times faster than (batSARank, powSARank, FRank, HRank) on DBLP and MAG, respectively.

In our tests we adopted a yearly update policy due the limitation of available time information. In practice our algorithms may bring more efficiency benefits since the ranking is usually more frequent, such that the data updates are smaller and the unaffected area is very likely much larger.

Exp-4: Impacts of parameters. In the last set of tests, we evaluated the impacts of time decaying factor σ, importance weighting factor λ, aggregating parameters α and β, and the TWPageRank. We fixed these parameters as well as Y_s to their default values, used the TWPageRank proposed in this work by default, and tested the PairAcc with the entire RECOM and PFCTN (*i.e.*, $T_i = +\infty$, $dif = 1$).

Exp-4.1. To evaluate the impacts of the time decaying factor



(a) AAN with RECOM (b) DBLP with RECOM (c) MAG with RECOM

(d) AAN with PFCTN (e) DBLP with PFCTN (f) MAG with PFCTN

Figure 7. Accuracy tests: varying aggregating parameters α and β

Table IV
ACCURACY TESTS USING DIFFERENT COMPONENTS WITH RECOM (ROWS 2–4) AND PFCTN (ROWS 5–7).

Datasets	**C**	**V**	**A**	**CV**	**CA**	**VA**	**CVA**
AAN	0.752	0.616	0.649	0.809	0.764	0.747	**0.810**
DBLP	0.735	0.581	0.640	0.784	0.749	0.729	**0.785**
MAG	0.635	0.534	0.553	0.697	0.673	0.648	**0.698**
AAN	0.785	0.557	0.761	0.849	0.866	0.771	**0.870**
DBLP	0.713	0.603	0.725	0.843	0.847	0.740	**0.856**
MAG	0.736	0.628	0.718	0.848	0.857	0.751	**0.874**

σ, we varied σ from -1.6 to -0.4. The results of PairAcc are reported in Figs. 5(d), 5(i) and 5(n).

When varying σ, the PairAcc of SARank is very stable on all datasets using both RECOM and PFCTN. Indeed, with RECOM and PFCTN, the PairAcc only varies (0.42%, 1.55%, 0.81%) and (1.26%, 0.96%, 1.16%) on (AAN, DBLP, MAG), respectively. The running time varies (11.3%, 8.6%) on average only on (DBLP, MAG), respectively.

Exp-4.2. To evaluate the impacts of importance weighting factor λ, we varied λ from 0 to 1. The results of PairAcc are reported in Figs. 5(e), 5(j) and 5(o). Note that parameter λ has no impacts on efficiency.

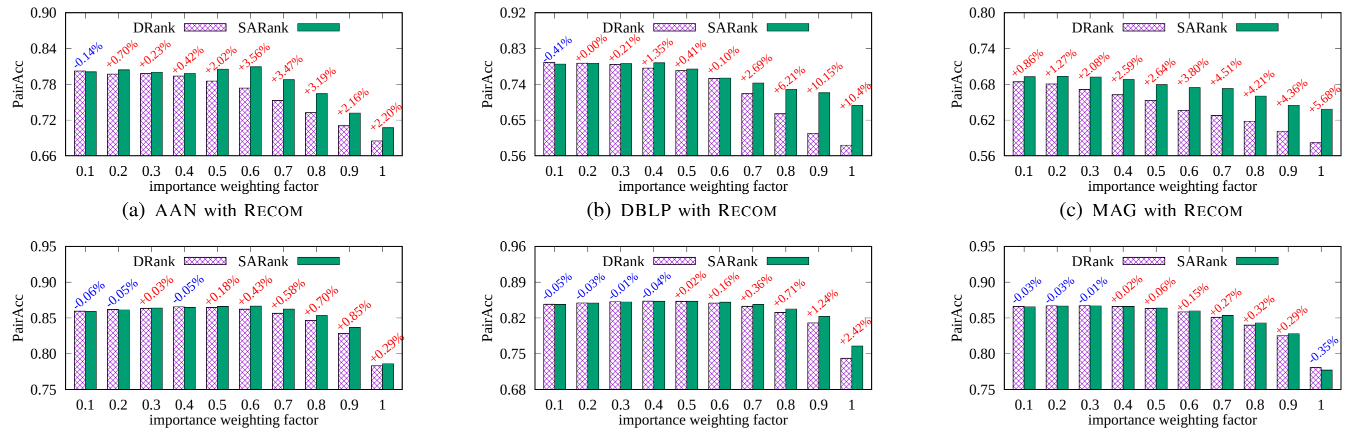
When varying λ, the PairAcc of SARank first increases and then decreases on all datasets with both PFCTN and RECOM, except on DBLP with RECOM. This result indicates that combining prestige and popularity generally produces more robust results than using either of prestige and popularity. Indeed, with RECOM and PFCTN, the PairAcc of combining prestige and popularity is (10.2%, 10.7%, 5.5%) and (8.0%, 8.7%, 9.0%) higher than using prestige alone, and is (1.2%, -0.1%, 1.0%) and (1.0%, 1.0%, 0.3%) higher than using popularity alone on (AAN, DBLP, MAG), respectively.

Exp-4.3. To evaluate the impacts of aggregating parameters α and β, we varied α and β at the granularity of 0.01. Again, parameters α and β have few impacts on efficiency. The results are reported in Fig. 7, where the parameters selected earlier and their corresponding PairAcc are marked with $*$.

When varying α and β, the PairAcc of SARank changes gently, as shown in Fig. 7. The optimal PairAcc is obtained within a single region, rather than a complex collection of optimal regions. Moreover, the PairAcc keeps at a high level within a certain (α, β) combination space around the optimal region, as shown in Fig. 7. Further, the optimal parameters on the same sets of ground-truth are very similar for (AAN, DBLP and MAG), indicating that the setting of α and β can be easily transferred across different datasets. To conclude, SARank is very robust to parameters α and β, and it is quite flexible for choosing proper values of parameters α and β.

Moreover, this enables to verify the effectiveness of importance assembling from different components, whose results are reported in Table IV, in which letters C, V and A stand for citation, venue and author components, respectively. The ranking based on all components consistently performs the best, using both RECOM and PFCTN, which justifies the use of importance assembling for ranking scholarly articles.

Exp-4.4. To evaluate the impacts of the proposed TWPageRank, we compared our approach SARank with an algorithm alternative (referred to as DRank) the same to SARank except exploiting exponentially decayed impact weights, *i.e.,* $w(u, v) = e^{\sigma(T_u - T_v)}$ in Eq. (1). To better understand the impacts, we varied the importance weighting factor λ from 0.1 to 1. Note that the ranking results are the same when $\lambda = 0$ due to the same popularity computation. The results are reported in Fig. 8, where the numbers represent the improvement of PairAcc by SARank over the one by DRank.



(a) AAN with RECOM (b) DBLP with RECOM (c) MAG with RECOM

(d) AAN with PFCTN (e) DBLP with PFCTN (f) MAG with PFCTN

Figure 8. Impacts of the TWPageRank on accuracy: varying importance weighting factor λ

When varying λ, the PairAcc of SARank is better than the one of DRank in most cases, which shows the superiority of the TWPageRank than exploiting exponentially decayed weights. The difference of PairAcc by the two algorithms is higher with RECOM than with PFCTN, since the two algorithms are using citation information to predict past and future citations with PFCTN. Moreover, algorithm SARank is consistently better than DRank when $0.5 \leq \lambda \leq 0.9$. The improvement decreases with the decrease of λ as the popularity dominates the ranking with small λ, and in some cases, DRank outperforms SARank. Overall, with RECOM and PFCTN, SARank improves the PairAcc over DRank by (1.78%, 3.07%, 3.20%) and (0.29%, 0.48%, 0.11%) on (AAN, DBLP, MAG) on average, respectively.

The TWPageRank has little impacts on efficiency, and the running time of the two algorithms only varies (6.34%, 4.83%) on (DBLP, MAG) on average, respectively.

Summary. From these tests we find the followings.

(1) Our model SARank is effective for ranking scholarly articles, which is consistently better than competitive methods in all tests. With RECOM and PFCTN, SARank improves PairAcc over (PRank, FRank, HRank) by (13.5%, 6.8%, 4.8%) and (12.0%, 3.0%, 3.2%) on AAN, (12.7%, 5.0%, 4.9%) and (14.0%, 6.5%, 4.6%) on DBLP, and (6.5%, 2.5%, 2.2%) and (13.4%, 6.0%, 2.4%) on MAG, on average, respectively.

(2) Our batch algorithm batSARank and incremental algorithm incSARank are also efficient. Our incremental algorithm incSARank is on average (1.7, 3.1, 2.8, 117) and (2.0, 3.0, 4.4, 245) times faster than (batSARank, powSARank, FRank, HRank) on the large DBLP and MAG, respectively.

(3) Our ranking model SARank introduces the time decaying factor σ, importance weighting factor λ and aggregating parameters α and β for the sake of practicability and flexibility in real-life applications, and, from our tests, SARank is very robust to these parameters. Moreover, the proposed TWPageRank is generally more effective than directly using exponentially decayed impact weights.

VI. RELATED WORK

Scholarly article ranking has shifted from citation count analysis [1], [12] to graph analysis [2], [3], [5], [6], [19]–[22], [33], [34]. Based on the information used, these methods are divided into four categories: (a) using the citation information only [1], [2], [12], [34], (b) using the citation and temporal information [19], [22], (c) using the citation information and other heterogeneous information, *e.g.,* authors and venues of articles [3], [5], [6], and (d) combining the citation, temporal and other heterogeneous information [20], [21], [33]. Our work belongs to the last category aiming at fully employing information available for scholarly article ranking. Moreover, recent work has also leveraged external data to improve the ranking quality, *e.g.,* using Knowledge Graph Embedding to better understand the meaning of research concepts [35], and has explored scientific journal ranking [36] and scholar ranking [37]. Different from these, our work ranks articles based on scholarly data only.

PageRank [25] and its extensions have been extensively used for citation analyses [7]. While PageRank equally propagates scores along outlinks, Weighted PageRank extends PageRank by distributing scores based on certain criteria such as popularity of pages [38] or authority of authors [39]. Both of them fail to capture the time-dependent characteristics, a key factor for scholarly article ranking. There has been work extending temporal information into PageRank, *e.g.,* exponential decayed weights [19], exponential decayed initial vector [22] and time-dependent weights based on co-authorship [40]. Different from previous work, the Time-Weighted PageRank of this work discriminately propagates scores in terms of citation statistics.

Dynamic algorithms have proven useful for various tasks by avoiding computing from scratch [41]. To our knowledge, little concern has been paid to dynamic scholarly article ranking except that [23] uses PageRank in dynamic citation networks. However, its solution is based on a strong and impractical assumption that there are no citations between articles in the same years. Further, although there exist several studies on incremental PageRank computation [42]–[44] and on incremental PageRank approximation [45], [46], they are not designed for scholarly article ranking. Different from previous work, we study scholarly article ranking in a dynamic environment in terms of the citation characteristics of scholarly articles, which has never been exploited before.

Ensemble methods use multiple learners to obtain better performance than could be obtained from a constituent learner alone [47]. In this work, we leverage importance assembling to produce better and more robust results for scholarly article ranking [18], [47], [48].

VII. Conclusions

We have proposed a model SARank for scholarly article ranking, which assembles the importance of article, venue and author entities. We have also proposed efficient batch and incremental algorithms for the computation of their importance, a combination of prestige and popularity. As shown by the experimental study, our approach is both effective and efficient for scholarly article ranking.

A couple of topics need further investigation. First, we are to clean scholarly data with external data sources and extend our model with affiliation and discipline information for further improving the quality of ranking. Second, we are to study distributed algorithms for importance computations, similar to [49] that computes PageRank in distributed scenarios.

References

[1] E. Garfield, "Citation analysis as a tool in journal evaluation," *Science*, vol. 178, no. 4060, pp. 471–479, 1972.

[2] P. Chen, H. Xie, S. Maslov, and S. Redner, "Finding scientific gems with google's pagerank algorithm," *J. Informetrics*, vol. 1, no. 1, pp. 8–15, 2007.

[3] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles, "Co-ranking authors and documents in a heterogeneous network," in *ICDM*, 2007.

[4] J. Shen, Z. Song, S. Li, Z. Tan, Y. Mao, L. Fu, L. Song, and X. Wang, "Modeling topic-level academic influence in scientific literatures," in *AAAI Workshop on Scholarly Big Data*, 2016.

[5] R. Liang and X. Jiang, "Scientific ranking over heterogeneous academic hypernetwork," in *AAAI*, 2016.

[6] X. Jiang, X. Sun, and H. Zhuge, "Towards an effective and unbiased ranking of scientific literature through mutual reinforcement," in *CIKM*, 2012.

[7] L. Waltman and E. Yan, *PageRank-Related Methods for Analyzing Citation Networks*. Springer, 2014, pp. 83–100.

[8] Z. Tan, C. Liu, Y. Mao, Y. Guo, J. Shen, and X. Wang, "Acemap: A novel approach towards displaying relationship among academic literatures," in *WWW,Companion Volume*, 2016.

[9] Semantic Scholar, *https://www.semanticscholar.org/*.

[10] Google Scholar, *https://scholar.google.com/*.

[11] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and K. Wang, "An overview of microsoft academic service (MAS) and applications," in *WWW*, 2015.

[12] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 46, pp. 16 569–16 572, 2005.

[13] C. C. Aggarwal and K. Subbian, "Evolutionary network analysis: A survey," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 10:1–10:36, 2014.

[14] S. Ma, J. Li, C. Hu, X. Lin, and J. Huai, "Big graph search: challenges and techniques," *FCS*, vol. 10, no. 3, pp. 387–398, 2016.

[15] D. Wang, C. Song, and A. Barabási, "Quantifying long-term scientific impact," *Science*, vol. 342, no. 6154, pp. 127–132, 2013.

[16] T. Chakraborty, S. Kumar, P. Goyal, N. Ganguly, and A. Mukherjee, "On the categorization of scientific citation profiles in computer science," *Commun. ACM*, vol. 58, no. 9, 2015.

[17] L. Bornmann and R. Mutz, "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references," *JASIST*, vol. 66, no. 11, pp. 2215–2222, 2015.

[18] A. D. Wade, K. Wang, Y. Sun, and A. Gulli, "Wsdm cup 2016: Entity ranking challenge," in *WSDM*, 2016.

[19] X. Li, B. Liu, and P. Yu, "Time sensitive ranking with application to publication search," in *ICDM*, 2008.

[20] Y. Wang, Y. Tong, and M. Zeng, "Ranking scientific articles by exploiting citations, authors, journals and time information," in *AAAI*, 2013.

[21] H. Sayyadi and L. Getoor, "Future rank: Ranking scientific articles by predicting their future pagerank," in *SDM*, 2009.

[22] D. Walker, H. Xie, K.-K. Yan, and S. Maslov, "Ranking scientific publications using a model of network traffic," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2007, no. 06, p. P06010, 2007.

[23] R. Ghosh, T. Kuo, C. Hsu, S. Lin, and K. Lerman, "Time-aware ranking in dynamic citation networks," in *ICDM Workshops*, 2011.

[24] M. Richardson, A. Prakash, and E. Brill, "Beyond pagerank: Machine learning for static ranking," in *WWW*, 2006.

[25] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, no. 1-7, pp. 107–117, 1998.

[26] Y. Dong, H. Ma, Z. Shen, and K. Wang, "A century of science: Globalization of scientific collaborations, citations, and innovations," in *KDD*, 2017.

[27] G. M. D. Corso, A. Gulli, and F. Romani, "Fast pagerank computation via a sparse linear system," *Internet Mathematics*, vol. 2, no. 3, pp. 251–273, 2005.

[28] P. Berkhin, "Survey: A survey on pagerank computing," *Internet Mathematics*, vol. 2, no. 1, pp. 73–120, 2005.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.

[30] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[31] A. Z. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. L. Wiener, "Graph structure in the web," *Computer Networks*, vol. 33, no. 1-6, pp. 309–320, 2000.

[32] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *KDD*, 2008.

[33] S. Wang, S. Xie, X. Zhang, Z. Li, P. S. Yu, and Y. He, "Coranking the future influence of multiobjects in bibliographic network through mutual reinforcement," *ACM TIST*, vol. 7, no. 4, pp. 64:1–64:28, 2016.

[34] M. K.-P. Ng, X. Li, and Y. Ye, "Multirank: Co-ranking for objects and relations in multi-relational data," in *KDD*, 2011.

[35] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *WWW*, 2017.

[36] M. Packalen and J. Bhattacharya, "Neophilia ranking of scientific journals," *Scientometrics*, vol. 110, no. 1, pp. 43–64, 2017.

[37] J. Zhang, Z. Ning, X. Bai, X. Kong, J. Zhou, and F. Xia, "Exploring time factors in measuring the scientific impact of scholars," *Scientometrics*, vol. 112, no. 3, pp. 1301–1321, 2017.

[38] W. Xing and A. Ghorbani, "Weighted pagerank algorithm," in *CNSR*, 2004.

[39] Y. Ding, "Applying weighted pagerank to author citation networks," *JASIST*, vol. 62, no. 2, pp. 236–245, 2011.

[40] D. Fiala, "Time-aware pagerank for bibliographic networks," *Journal of Informetrics*, vol. 6, no. 3, pp. 370 – 388, 2012.

[41] G. Ramalingam and T. W. Reps, "A categorized bibliography on incremental computation," in *POPL*, 1993.

[42] P. K. Desikan, N. Pathak, J. Srivastava, and V. Kumar, "Incremental page rank computation on evolving graphs," in *WWW, Special interest tracks and posters*, 2005.

[43] S. Abiteboul, M. Preda, and G. Cobena, "Adaptive on-line page importance computation," in *WWW*, 2003.

[44] Y. Wu and L. Raschid, "Approxrank: Estimating rank for a subgraph," in *ICDE*, 2009.

[45] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *PVLDB*, vol. 4, no. 3, pp. 173–184, 2010.

[46] B. Bahmani, R. Kumar, M. Mahdian, and E. Upfal, "Pagerank on an evolving graph," in *KDD*, 2012.

[47] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 2012.

[48] L. Duan, C. Aggarwal, S. Ma, R. Hu, and J. Huai, "Scaling up link prediction with ensembles," in *WSDM*, 2016.

[49] Y. Zhu, S. Ye, and X. Li, "Distributed pagerank computation based on iterative aggregation-disaggregation methods," in *CIKM*, 2005.