Front. Comput. Sci., 2016, 10(1): 113–123
DOI 10.1007/s11704-015-4570-7

RESEARCH ARTICLE

Accuracy estimation of link-based similarity measures and its application

Yinglong ZHANG[1,3], Cuiping LI (✉)[2], Chengwang XIE[1,3], Hong CHEN[2]

1 School of Software, East China Jiaotong University, Nanchang 330045, China
2 Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education,
and Department of Computer Science, Renmin University of China, Beijing 100872, China
3 Intelligent Optimization and Information Processing Laboratory, East China Jiaotong University,
Nanchang 330013, China

Abstract Link-based similarity measures play a significant role in many graph based applications. Consequently, measuring node similarity in a graph is a fundamental problem of graph data mining. Personalized PageRank (PPR) and SimRank (SR) have emerged as the most popular and influential link-based similarity measures. Recently, a novel link-based similarity measure, penetrating rank (P-Rank), which enriches SR, was proposed. In practice, PPR, SR and P-Rank scores are calculated by iterative methods. As the number of iterations increases so does the overhead of the calculation. The ideal solution is that computing similarity within the minimum number of iterations is sufficient to guarantee a desired accuracy. However, the existing upper bounds are too coarse to be useful in general. Therefore, we focus on designing an accurate and tight upper bounds for PPR, SR, and P-Rank in the paper. Our upper bounds are designed based on the following intuition: the smaller the difference between the two consecutive iteration steps is, the smaller the difference between the theoretical and iterative similarity scores becomes. Furthermore, we demonstrate the effectiveness of our upper bounds in the scenario of top-*k* similar nodes queries, where our upper bounds helps accelerate the speed of the query. We also run a comprehensive set of experiments on real world data sets to verify the effectiveness and efficiency of our upper bounds.

Received December 15, 2014; accepted February 13, 2015

E-mail: cuiping_li@263.net

Keywords personalized PageRank, SimRank, P-Rank, upper bound

1 Introduction

On the Internet, graphs are ubiquitous; the use of graphs in the Web, social networks, bibliographic graphs, and entity-relationship graphs, calls for solutions to measure similarity between nodes. Measures of similarity between objects play a significant role in many graph applications, e.g., recommendation systems [1], link prediction [2], fraud detection [3], and collaborative filtering [4].

Many link-based similarity measures have been proposed, such as personalized PageRank (PPR) [5], SimRank (SR) [6], hitting time [7] and commute time [8]. Among them, both PPR and SR have emerged as the most popular and influential link-based similarity measures due to their effectiveness and solid theoretical foundation. Recently, a novel link-based similarity measure, penetrating rank (P-Rank) [9], was proposed. P-Rank enriches SR: it measures node similarity considering both the in- and out-link relationships of nodes, where SR neglects the effect of out-link relationships.

Although iterative similarity scores of PPR, SR, and P-Rank are convergent [5, 6, 9], in practice the corresponding computations naturally involve performing a finite number of iterations. PPR, SR, and P-Rank computations are time-consuming. As the number of iterations increases, the computations cause significant overhead, especially on a large graph. In Ref. [10], given a graph which consists of 10 000 nodes, Lizorkin et al run the original iterative SR on a 2.1 GHz Intel Pentium processor with 1 GB RAM. After 5 iterations, it took 46 hours and 5 minutes for the algorithm to obtain all node pairs similarities. Because P-Rank enriches SR, in theory the overhead of P-Rank computations is much higher than the that of SR.

However, the existing upper bounds are too coarse to be useful in general. It has been advised that to choose the decay factor $c = 0.8$ and total iterations $K = 5$ to compute iterative SR similarity [6], and, according to proposition 1 in Ref. [10], the corresponding difference between theoretical and computed similarity scores is 0.26. Based on the Lemma 2 in Ref. [11], the difference between theoretical and iterative PPR scores is also 0.26, when $c = 0.8$ and the total iterations $K = 5$. The existing upper bounds of PPR and SR are relatively large because the interval of the theoretical scores is [0,1]. For P-Rank, there is not much work that focuses on accuracy estimation of its iterative computation.

Accordingly, an accurate difference between iterative similarity scores and theoretical scores remains an open question. The ideal solution is that computing similarity within the minimum number of iterations is sufficient to guarantee a desired accuracy.

At the *i*th iteration, if the iterative score is P_i and the difference between theoretical and computed similarity score is $P - P_i \leqslant \delta_i$, then the corresponding upper bound is $P_i + \delta_i$, and symmetrically, if the upper bound of P_i is $P_i + \delta_i$, then the difference is δ_i. Given this relationship we use the terms upper bound and difference interchangeably.

In summary, it is important to design an accurate and tight upper bounds, both in theory and in practice. Given a desired accuracy, we should terminate the iteration as soon as possible to reduce overhead by leveraging the tight upper bound. Furthermore we can accelerate graph-based queries such as link-based similarity join [11] and top-*k* similarity search by utilizing a tight upper bound.

We say that the difference between iterative and theoretical similarity scores is good if the following properties hold:

- Accurate: the difference is very close to the true difference. For example, if the difference between theoretical and computed similarity score δ_i and δ_j are respectively estimated by methods A and B, if $\delta_i < \delta_j$, we say that δ_i is closer to the true difference than δ_j's and δ_i is an accurate difference.
- Fast: we can obtain the difference quickly with minimal computation cost.

So, if the above holds, our differences are accurate and can be obtained quickly.

In this paper we focus on designing an accurate and tight upper bounds for PPR, SR, and P-Rank. Our upper bounds are designed based on following the intuition that the smaller the difference between two consecutive iteration steps is, the smaller the difference between iterative and theoretical similarity scores becomes.

This article is an extended version of our conference paper [12]. In this article, we add two types upper bound for P-Rank. First we give a concise upper bound of P-Rank. Then we propose a second upper bound based on key iteration intuition mentioned above. The two upper bounds are proved mathematically.

In Section 2, we introduce the necessary notations and formulas. In Section 3, we propose an accurate and tight upper bounds of PPR, SR, and P-Rank. In Section 4, we tailor our upper bounds to accelerate the top-k similar nodes query. In Section 5 experimental results are presented. Section 6 gives an overview of the related work. Our conclusion is given in Section 7.

2 Preliminary

Given a directed graph $G = (V, E)$ where nodes in V represent objects, and edges in E represent relationships between objects. For any $v \in V$, Set $I(v)$ and $O(v)$ respectively denote in-neighbors and out-neighbors of v. $I_i(v)$ or $O_j(v)$ is an individual member of $I(v)$, for $1 \leqslant i \leqslant |I(v)|$, or of $O(v)$, for $1 \leqslant j \leqslant |O(v)|$.

2.1 PPR

The Web can be viewed as a directed graph of pages connected by hyperlinks. A random web surfer starts from an arbitrary page and simply keeps clicking on successive links at random, bouncing from page to page. Like PageRank, PPR is the steady-state probabilities of random walks; at each step, a web surfer randomly walks along an out link with probability c, and with probability 1-c return to a random node of the set of preferred nodes. If the preferred set contains only one node, PPR actually is random walk with restart (RWR). RWR is a special case of PPR. In this paper we only consider the situation that the preferred set contains one node (a query node).

According to [5, 11], the equation of PPR is

$$r(q, v) = (1 - c) \sum_{\tau:q \to v} P(\tau) c^{l(\tau)}, \tag{1}$$

where τ is the unidirectional path from q to v: $(q, w_1, \ldots, w_n, v)$, $l(\tau)$ is the length of the path τ, $P(\tau) = \frac{1}{|O(q)|} \prod_{i=1}^{n} \frac{1}{|O(w_i)|}$ is the probability of traversing the τ, and $r(q, v)$ is the similarity between q and v from q's personalized view. In practice

$$r_k(q, v) = (1 - c) \sum_{\substack{\tau:q \sim v \\ l(\tau) \leqslant k}} P(\tau) c^{l(\tau)}, \tag{2}$$

is used to estimate $r(q, v)$.

2.2 SR

SR measures the similarity of nodes based on following human intuition: "two objects are similar if they are related to similar objects" [6]. So the SR score (a, b) is the average SR score between in-neighbors of a and in-neighbors of b:

$$s(a, b) = \begin{cases} 1, & \text{if } a = b; \\ \frac{c \sum_i^{|I(a)|} \sum_i^{|I(b)|} s(I_i(a), I_j(b))}{|I(a)||I(b)|}, & I(a) \text{ and } I(b) \neq \emptyset; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Correspondingly, as shown in [6] the iterative formula is:

$$s_{k+1}(a, b) = \frac{c}{|I(a)||I(b)|} \sum_{i}^{|I(a)} \sum_{j}^{|I(b)|} s_k(I_i(a), I_j(b)),$$
$$k = 0, 1, 2, \ldots \tag{4}$$

The SR score measures how soon two random surfers are expected to meet at the same node if they started at nodes a and b and randomly walked the graph backwards [6]. According to Ref. [6], the formula of SR can be written as follows:

$$s(a, b) = \sum_{\tau:(a,b) \to (x,x)} P(\tau) c^{l(\tau)}, \tag{5}$$

where τ is a tour (paths may have cycles) along which two random suffers walk backwards starting at nodes a and b, respectively, until they first for the first and only time at any node x, $l(\tau)$ is the length of tour τ.

Based on Ref. [13], the corresponding iterative formula is:

$$s_k(a, b) = \sum_{\substack{\tau:(a,b) \to (x,x) \\ l(\tau) \leqslant k}} P(\tau) c^{l(\tau)}. \tag{6}$$

2.3 P-Rank

In contrast to SR, P-Rank considers both in- and out-link relationships of node pairs. As discussed in Ref. [9], the key concepts of of P-Rank are

1) Two objects are similar if they are referenced by similar objects.
2) Two objects are similar if they reference similar objects

P-Rank is defined as follows, when $a \neq b$

$$w(a, b) = \delta \times \frac{c}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} w(I_i(a), I_j(b)) + (1 - \delta)$$
$$\times \frac{c}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} w(O_i(a), O_j(b)). \tag{7}$$

otherwise,

$$w(a, b) = 1. \tag{8}$$

In the above equation, $\delta \in [0, 1]$ is used to balance the relative weight of in- and out-link directions, and $c \in [0, 1]$ is a damping factor.

The iterative form of P-Rank is as follows

$$w_0(a, b) = \begin{cases} 0, & \text{if } a \neq b; \\ 1, & \text{if } a = b, \end{cases} \tag{9}$$

and

$$w_{k+1}(a, b) = \delta \times \frac{c}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} w_k(I_i(a), I_j(b)) + (1 - \delta)$$
$$\times \frac{c}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} w_k(O_i(a), O_j(b)), \tag{10}$$

where $w_k(a, b)$ denotes the P-Rank score between a and b on iteration k, for $a \neq b$ and $w_k(a, b) = 1$ for $a = b$.

2.4 Problem of top-*k* similar nodes query

We focus on designing an accurate and tight upper bounds of PPR, SR, and P-Rank. By leveraging the tight upper bound, we can terminate the iteration as soon as possible to reduce overhead. Furthermore, we also can accelerate graph-based queries by utilizing a tighter upper bound. Here we give the definition of the top-*k* query, using which we will later evaluate the performance of our approach.

In this paper, we use $P(a, b)$ and $P_w(a, b)$ to respectively denote the similarity score and corresponding iterative score of (a,b) for any one of these measures: PPR, SR, and P-Rank.

Problem statement (top-*k* similar nodes query) Given a query node q, a required number of results k, and ϵ, the result of query, the top-*k* similarity nodes of q, is $T_k(q) = \{t_1, \ldots, t_k\}$ if similarity score $P_w(q, t_i) \geqslant P_w(q, t)$ $(\forall t \in V(G(V))/T_k(q))$ on the graph $G(V)$.

3 Bounds of link-based similarity measures

The existing upper bounds of PPR and SR are too coarse to be useful in general. For P-Rank, not much work focuses on accuracy estimation of its iterative computation. In this section we introduce a novel upper bounds of these link-based measures. Our upper bounds are designed based on the following intuition: the smaller the difference between the two consecutive iteration steps is, the smaller the difference between the theoretical and iterative similarity scores becomes.

The intuition is based on the following analysis. Given a specific node pair (a, b), its iterative values of PPR (or SR or P-Rank) consist of a convergent sequence [5, 6, 9]. In other words, with the increase in number of iterations k, the corresponding iterative value gets very close to the limit: theoretical value of similarity. As a result, two consecutive iteration step results are very close to each other. This coincides with well-known Cauchy sequence[1]: A sequence $\{a_n\}$ of real numbers has a finite limit if and only if for every $\epsilon > 0$ there is an N such that $|a_n - a_m| < \epsilon$ for every $n, m \geqslant N$. Therefore, the intuition is reasonable.

In this section, these novel upper bounds are strictly proven. The procedures of the proofs also shed light on the essence of these link-based similarity measures.

3.1 Bounding of PPR

Along any path τ from q to v: $(q, O_i(q), \ldots, w_n, v)$, a surfer walks one step beforehand from node q to its out-neighbor $O_i(q)$ with equal probability $\dfrac{1}{|O(q)|}$, and $P(\tau) = P(\tau')/|O(q)|$ where τ' is the path: $(O_i(q), \ldots, w_n, v)$, therefore Eq. (1) can be transformed as:

$$\begin{aligned} r(q,v) &= (1-c)\sum_{\tau:q\to v} P(\tau)c^{l(\tau)} \\ &= \frac{(1-c)c}{|O(q)|}\sum_{i=1}^{|O(q)|}\sum_{\tau':O_i(q)\to v} P(\tau')c^{l(\tau')} \\ &= \frac{c}{|O(q)|}\sum_{i=1}^{|O(q)|} r(O_i(q), v). \end{aligned}$$

Similar, we have:

$$r_{k+1}(q,v) = \frac{c}{|O(q)|}\sum_{i=1}^{|O(q)|} r_k(O_i(q), v). \tag{11}$$

At mth iteration, let $\delta_m = \max\limits_{\forall a\in V, b\in V}\{(r_m(a,b) - r_{m-1}(a,b))\}$, we have following theorem:

Theorem 1 The difference between theoretical and iterative PPR scores is

$$r(q,v) - r_m(q,v) \leqslant \delta_m \frac{c}{1-c}. \tag{12}$$

Proof According to Eq. (2),

$$r_{m+1}(q,v) - r_m(q,v) = (1-c)\sum_{\substack{t:q\sim v \\ l(t)=m+1}} P(\tau)c^{l(\tau)}.$$

Thus

$$\begin{aligned} r(q,v) - r_m(q,v) &= (1-c)\sum_{\substack{t:q\sim v \\ l(t)=m+1}}^{\infty} P(\tau)c^{l(\tau)} \\ &= \sum_{i=1}^{\infty}(r_{m+i}(q,v) - r_{m+i-1}(q,v)). \end{aligned}$$

Based on Eq. (11) and $\delta_m = \max_{a\in V, b\in V}\{(r_m(a,b) - r_{m-1}(a,b))\}, \forall a, b \in V$

$$\begin{aligned} & r_{m+1}(a,b) - r_m(a,b) \\ &= \frac{c}{|O(a)|}\sum_{i=1}^{|O(a)|}(r_m(O_i(a),v) - r_{m-1}(O_i(a),v)) \\ &\leqslant \frac{c}{|O(a)|}\sum_{i=1}^{|O(a)|}\delta_m \\ &= c\delta_m. \end{aligned}$$

Likewise, $r_{m+k}(a,b) - r_{m+k-1}(a,b) \leqslant c^k\delta_m$. Therefore

$$\begin{aligned} & r(q,v) - r_m(q,v) \\ &= \sum_{i=1}^{\infty}(r_{m+i}(q,v) - r_{m+i-1}(q,v)) \\ &\leqslant \sum_{i=1}^{\infty} c^i\delta_m = \delta_m\frac{c - c^{\infty}}{1-c} \\ &= \delta_m\frac{c}{1-c}. \end{aligned}$$

Theorem 1 gives the lower and upper bounds of PPR at the mth iteration: $r_m(q,v) \leqslant r(q,v) \leqslant r_m(q,v) + \delta_m c/1 - c$. At the $(m+k)$th iteration, we update δ_{m+k} as follows $\delta_{m+k} = \min(\max\limits_{a\in V, b\in V}\{(r_{m+k}(a,b) - r_{m+k-1}(a,b))\}, \delta_{m+k-1})$.

Lemma 2 in Ref. [11] gives an upper bound of PPR, c^{m+1}, at the mth iteration. The following proposition states our upper bound is better (lower) than that in Ref. [11].

Proposition 1 At mth iteration, $\delta_m\dfrac{c}{1-c} \leqslant c^{m+1}$.

Proof

$$\delta_m \leqslant \max_{a\in V, b\in V}\{(r_m(a,b) - r_{m-1}(a,b))\}$$

[1] http://www. encyclopediaofmath.org/index.php?title=Cauchy_criteria&oldid=30908

$$= \max_{a\in V, b\in V}\{(1-c)\sum_{\substack{t:a\sim b\\ l(t)=m}} P(\tau)c^{l(t)}\}$$

$$\leqslant (1-c)c^m$$

because $\sum_{\substack{t:a\sim b\\ l(t)=m}} P(\tau) \leqslant 1$.

Proposition 1 guarantees that our upper bound is superior to that in Ref. [11] in theory. Our upper bound reduces dramatically as the number of iterations increases because $\delta_m \frac{c}{1-c} \leqslant c^{m+1}$.

3.2 Bounding of SR

At the mth iteration, let $\delta_m = \max\limits_{\forall a\in V, b\in V}\{s_m(a,b) - s_{m-1}(a,b)\}$, we have following theorem:

Theorem 2 The difference between theoretical and iterative SR scores is

$$s(a,b) - s_m(a,b) \leqslant \delta_m \frac{c}{1-c}. \tag{13}$$

Proof Based on Eq. (4),

$$\begin{aligned}
&S_{m+1}(a,b) - S_m(a,b)\\
&= \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}(S_m(I_i(a), I_j(b)) - S_{m-1}(I_i(a), I_j(b)))\\
&\leqslant \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}\delta_m = c\delta_m
\end{aligned}$$

as $\delta_m = \max\{s_m(a,b) - s_{m-1}(a,b)\}$ ($\forall a, b \in V$). Accordingly,

$$\begin{aligned}
&S_{m+2}(a,b) - S_{m+1}(a,b)\\
&= \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}(S_{m+1}(I_i(a), I_j(b)) - S_m(I_i(a), I_j(b)))\\
&\leqslant \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}c\delta_m = c^2\delta_m
\end{aligned}$$

likewise: $s_{m+k}(a,b) - s_{m+k-1}(a,b) \leqslant c^k\delta_m$.

On the other hand, according to Eqs. (6) and (5),

$$s_{m+1}(a,b) - s_m(a,b) = \sum_{\substack{\tau:(a,b)\to(x,x)\\ l(\tau)=m+1}} P(\tau)c^{l(\tau)},$$

and $s(a,b) - s_m(a,b) = \sum\limits_{\substack{\tau:(a,b)\to(x,x)\\ l(\tau)=m+1}}^{\infty} P(\tau)c^{l(\tau)}$

$$= \sum_{k=1}^{\infty} s_{m+k}(a,b) - s_{m+k-1}(a,b)$$

$$= \sum_{k=1}^{\infty} c^k\delta_m = \delta_m \sum_{k=1}^{\infty}\frac{c - c^{\infty}}{1-c} = \delta_m\frac{c}{1-c}.$$

Theorem 2 gives the lower and upper bounds of SR at the mth iteration. As soon as the difference $\delta_m \frac{c}{1-c}$ satisfies the given precision, we can stop. According to the result of experiments, when $m \geqslant 3$, the difference is largely less than the difference c^{m+1}, which is proposed in Ref. [10], although we can not prove it in theory. In order to obtain better result, $\min\{\delta_m \frac{c}{1-c}, c^{m+1}\}$ is the same as the SR difference when $m < 3$.

3.3 Bounding of P-Rank

First we give a concise upper bound of P-Rank, inspired by Ref. [14]. Then we propose another upper bound based on our intuition.

Theorem 3 The difference between theoretical and iterative P-Rank scores is

$$w(a,b) - w_m(a,b) \leqslant c^{m+1}. \tag{14}$$

Proof For the general case $a \neq b$, we prove the theorem by mathematical induction.

- Induction Basis. We first prove that Eq. (14) holds when $m = 0$: Due to the definition of P-Rank, $w_0(a,b) = 0$, and

$$\begin{aligned}
&w(a,b) - w_m(a,b) = w(a,b)\\
&= \delta\frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} w(I_i(a), I_j(b))\\
&\quad +(1-\delta)\frac{c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|} w(O_i(a), O_j(b))\\
&\leqslant \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} 1 + (1-\delta)\frac{c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|} 1\\
&= \delta c + (1-\delta)c = c
\end{aligned}$$

- Inductive step. Assume that Eq. (14) holds for m for all node pairs, then we prove that Eq. (14) also holds for $(m+1)$:

$$\begin{aligned}
&w(a,b) - w_{m+1}(a,b)\\
&= \delta\frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} w(I_i(a), I_j(b))\\
&\quad +(1-\delta)\frac{c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|} w(O_i(a), O_j(b))\\
&\quad -[\frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} w_m(I_i(a), I_j(b))\\
&\quad +(1-\delta)\frac{c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|} w_m(O_i(a), O_j(b))]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\delta c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}[w(I_i(a), I_j(b)) - w_m(I_i(a), I_j(b))]\\
&\quad + \frac{(1-\delta)c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|}[w(O_i(a), O_j(b))\\
&\quad - w_m(O_i(a), O_j(b))]\\
&\leqslant \frac{\delta c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}c^{k+1} + \frac{(1-\delta)c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|}c^{k+1}\\
&= \delta c^{k+1} + (1-\delta)c^{k+1} = c^{k+1}.
\end{aligned}
$$

Theorem 3 gives a concise upper bound of P-Rank. We now propose a further upper bound, based on our aforementioned intuition.

At mth iteration, let $\delta_m = \max_{\forall a\in V, b\in V}\{w_m(a,b) - w_{m-1}(a,b)\}$, we have following theorem:

Theorem 4 The difference between theoretical and iterative P-Rank scores is

$$w(a,b) - w_m(a,b) \leqslant \delta_m \frac{c}{1-c}. \tag{15}$$

Proof From Eq. (10),

$$
\begin{aligned}
&w_{k+1} - w_k\\
&= \delta \times \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}(w_k(I_i(a), I_j(b)) - w_{k-1}(I_i(a), I_j(b)))\\
&\quad +(1-\delta)\times\frac{c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|}\\
&\quad \times(w_k(O_i(a), O_j(b)) - w_{k-1}(O_i(a), O_j(b)))\\
&\leqslant \delta \times \frac{c}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}\delta_m\\
&\quad +(1-\delta)\times\frac{c}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|}\delta_k\\
&= \delta\times c\times\delta_k + (1-\delta)\times c\times\delta_k\\
&= c\delta_k = \delta_{k+1}
\end{aligned}
$$

Similarly, $w_{k+n}(a,b) - w_{k+n-1}(a,b) \leqslant \delta_{k+n} = c^n\delta_k$

According to [9], $w(a,b) = lim_{n\to\infty}w_k(a,b)$. Thus $w(a,b) - w_k(a,b) = \sum_{m=1}^{\infty} w_{m+k}(a,b) - w_{m+k-1}(a,b)) = \sum_{k=1}^{\infty} c^k\delta_k = \frac{c}{1-c}\delta_k$

In this section, the three novel upper bounds are obtained based on based our aforementioned intuition. For fully digesting and understanding aforementioned results, we further give following explanation. When δ_m, which is the maximum value among the differences between the two consecutive iteration step results, is very small, it means that previous iterative value is very close to the theoretic value according to the theory of Cauchy sequence. Observing Eqs. (4), (10) and (11), we find that the current iterative value of the three measures is the weighted sum of some last iterative values. Therefore the current iterative value is close to the theoretical value when the δ_m is small enough.

3.4 Obtain upper bounds

We do not need to spend extra overhead to obtain our upper bounds by incrementally updating similarity scores.

Equation (2) can be rewritten as

$$
\begin{aligned}
r_{k+1}(q,v) &= (1-c)\sum_{\substack{\tau:q\sim v\\ l(\tau)\leqslant k+1}} P(\tau)c^{l(\tau)}\\
&= r_k(q,v) + (1-c)\sum_{\substack{\tau:q\sim v\\ l(\tau)=k+1}} P(\tau)c^{l(\tau)}.
\end{aligned}
\tag{16}
$$

Likewise, Eq. (6) can be transformed into

$$s_{k+1}(a,b) = s_k(a,b) + \sum_{\substack{\tau:(a,b)\to(x,x)\\ l(\tau)=k+1}} P(\tau)c^{l(\tau)}. \tag{17}$$

The above two equations state that the current similarity score is the sum of previous score plus the increment. At each iteration we actually compute the increment to obtain the similarity score. And δ_m in Eq. (12) (or Eq. (13)) is the maximal increment. Optimization of PPR in Ref. [15] and SR in [13] are based on Eq. (16) and Eq. (17) respectively.

For P-Rank, the concise upper bound in Eq. (14) can be easily obtained without any overhead. Furthermore, the optimization technologies in Ref. [13] can be applied to compute the P-Rank scores. The corresponding maximal increment is the δ_m in Eq. (15).

Consequently, we do not need to spend extra overhead to obtain upper bounds.

4 Top-*k* similar nodes query

In this section, we demonstrate the effectiveness of our novel upper bounds in the scenario of top-k similar nodes query.

Observe from Eqs. (2) and (5) that $P(a,b)$ involves an infinite number of random walks (similar to SR, P-Rank also involves an infinite number of random walks). Consequently, it is infeasible to achieve an accurate $P(a,b)$. It is effective to compute $P_w(a,b)$ instead:

$$|P(a,b) - P_w(a,b)| \leqslant \epsilon, \tag{18}$$

where ϵ controls the accuracy of $P_w(a,b)$ in estimating $P(a,b)$, and w is the minimum value that satisfies the inequality.

Algorithm 1 Top-k similar nodes query

```
Input Graph $g$, $c$, $q$, $k$, $\varepsilon$
Output top-$k$ lists of $q$
Set $pathProb \leftarrow 1.0$
push pair $(v, pathProb)$ into queue $que$
while $!obtained$ do
    $(currentNode, pathProb) \leftarrow que.front()$
    $que.pop()$
    If $currentNode$ $! = -1$ then
        // the flag –1 indicates the current iteration is finished
        foreach a neighbors $j$ of $currentNode$ do
            $queTemp[j] \leftarrow queTemp[j] + \frac{pathProb}{OutDegree}$
            // walk one step to obtain probability of path from the query
               node q to the neighbor j
    else
        $i \leftarrow i + 1$
        foreach element $j$ of $queTemp$ do
            // Eq. (16):
            $rwrScore[j] \leftarrow rwrScore[j] + queTemp[j] \times (1 - c) \times c^i$
            $\delta \leftarrow \delta + queTemp[j] \times (1 - c) \times c^i$
            push $(j, queTemp[j])$ into $que$
        $\varepsilon' \leftarrow \delta / queTemp.size()$    // average local difference
        $\varepsilon' \leftarrow \varepsilon' \times c/(1 - c)$    // Upper bound Eq. (12)
        sort $rwrScore$ to obtain top k nodes
        // $\widehat{P(q,v)}(\forall v \in V(G(V))/T_k(q))$:
        if $T_k > \widehat{P(q,v)}$ or $\varepsilon' < \varepsilon$ then
            $obtained$ ←True
        push $(-1, pathProb)$ into $que$
        clear $queTemp$
        $\delta \leftarrow 0$
        continue
return $result$
```

General framework of top-k similar nodes query: Starting at a query node q, we do a breadth-first traverse to visit remaining nodes. At the mth iteration, when a node v is visited, we compute $P_m(q, v)$. After the mth iteration, we obtain its upper bound value $\widehat{P_m(q, v)}$ and ϵ_m (ϵ_m is the difference between theoretical and iterative similarity scores at the current iteration), then find a set of k nodes with the highest scores of lower bounds. Let T_k be the kth largest score. We terminate the query and obtain the final result of the top-k query if one of following conditions is true:

- $\epsilon_m \leqslant \epsilon$
- $T_k \geqslant \widehat{P_m(q, t)}$ ($\forall t \in V(G(V))/T_k(q)$).

With the help of our upper bound, we can obtain top-k nodes via local expansion around the query node. The local top-k similarity search method avoids accessing the whole graph.

The local top-k similar nodes search method effectively handles similarity search because it does not need to access the whole graph, especially, when the graph is very large. While, the upper bounds, are obtained based on the whole graph. Therefore, the upper bounds cannot be directly applied to the top-k query when only the local information of the query node is available. Furthermore, we tailor upper bounds based on theory mentioned in Section 3.

We customize upper bounds based on local information. At each iteration, we obtain the similarity scores between the query node and accessed nodes, which belong to the neighborhood of the query node. The average difference between the two consecutive iteration step results can be used to estimate δ_m in Eqs. (12), (13), and (15). As a result, we obtain variants of the upper bounds based on the local information. Although the accuracy of our three upper bounds cannot be theoretically proven, they are reasonable: the average difference is very close to the true difference due to the principle of locality.

Algorithm 1 is a top-k similar nodes query method based on PPR. A top-k method based on SR or P-Rank is similar to the Algorithm 1 in Ref. [16] and is not listed in this paper.

It is worth mentioning that although top-k queries are exploited to demonstrate effectiveness of the upper bounds, they are not our target. From the above analysis, the upper bounds play a key role in the algorithm. The upper bounds accelerate the query speed and avoid accessing the whole graph.

5 Experiments

We implemented all experiments on a PC with an I3-550 CPU, 4 G main memory, running Windows 7 64 bit operating system. All code is written in C++. In the experiment, the damping factor c = 0.8; for P-Rank, the relative weight λ is set to 0.5; $\epsilon = 10^6$.

The data sets used in the experiments are shown in Table 1. Cora[2] is a citation graph. Graphs FaceBook, Hamster (social graph), and Subelj (E-road network) can be visited at KONECT[3]. The remaining data can be visited at SNAP[4]. For the first 5 data sets in Table 1, we use their maximum graph components instead of the original graphs (the corresponding information in Table 1 is that of their maximum components).

The state-of-the-art upper bounds, PPR upper bound in Ref. [11] and SR upper bound in Ref. [10], are used as **base-**

[2] http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html

[3] http://konect.uni-koblenz.de/networks/

[4] http://snap.stanford.edu/data/index.html

Table 1 Data sets

	Cora	FaceBook	Hamster	Subelj	P2P-Gnutella06	Ego-Twitter	Web-Stanford
Nodes	2 485	2 887	1 999	1 039	8 717	81 306	281 903
Edges	5 209	5 388	31 676	2 484	31 525	1 768 149	2 312 497

lines to compare with our upper bounds in experiments. For P-Rank, Eq. (14) is used as the baseline to compare with the upper bound in Eq. (15).

When δ_m is estimated by the average of top-100 largest differences between the two consecutive iteration step results, the corresponding upper bounds are denoted as approximate and achieve a very high precision ($\geqslant 99\%$) in the scenario of top-k similar nodes query.

Figures 1–6 show the results of our accurate and approximate upper bounds compared with the baselines on 5 real data sets. It is worth mentioning that P-Rank only applies to a directed graph. Therefore the upper bound of P-Rank is only tested on the three directed graphs: P2P, FaceBook, and Cora. When $m \geqslant 3$, our SR difference is mostly less than the baseline although the rate of SR convergence is different on different real data sets. Our approximate PPR upper bound is

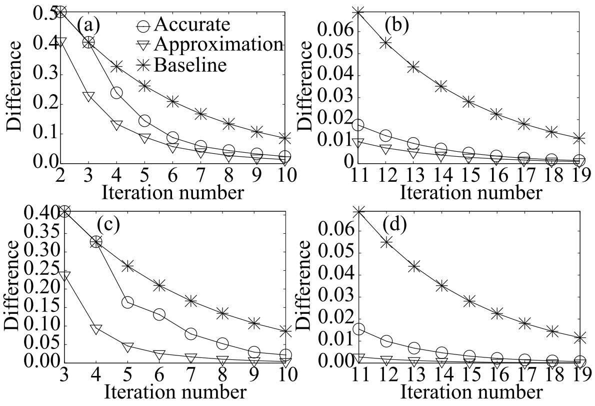

Fig. 1 Our SR difference vs. Baseline difference with iteration number (a) < 11 on Subelj; (b) > 10 on Subelj; (c) < 11 on Cora; (d) > 10 on Cora

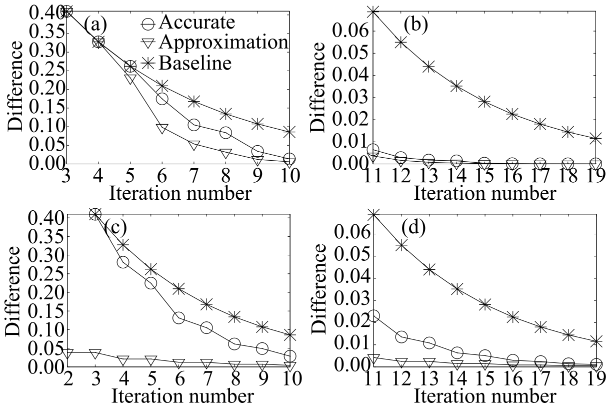

Fig. 2 Our SR difference vs. Baseline difference with iteration number (a) < 11 on P2P; (b) > 10 on P2P; (c) < 11 on Facebook; (d) > 10 on Facebook

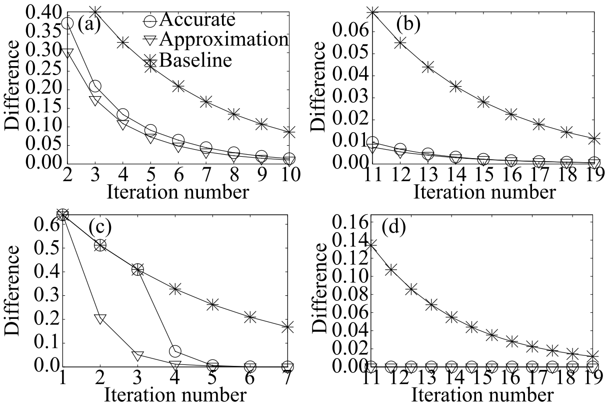

Fig. 3 Our difference vs. Baseline difference with iteration number (a) < 11 on Petster-hamster for SR; (b) > 10 on Petster-hamster for SR; (c) < 8 on P2P for PPR; (d) > 7 on P2P for PPR

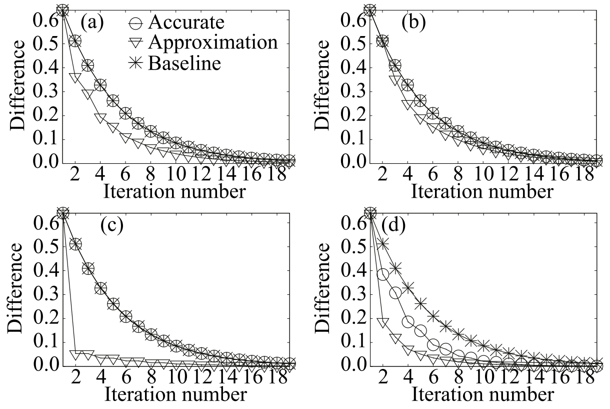

Fig. 4 Our PPR difference vs. Baseline difference. (a) Subelj; (b) Cora; (c) Facebook; (d) Petster-hamster

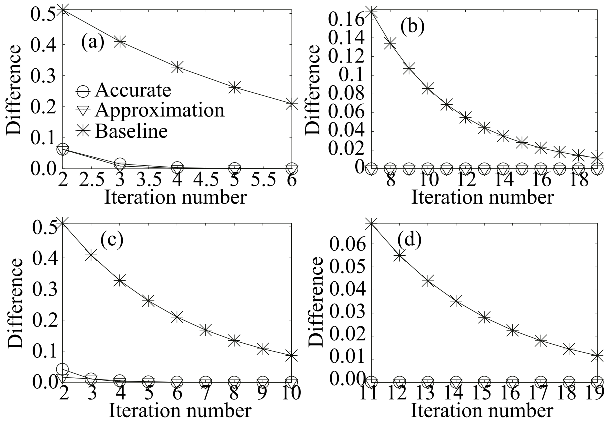

Fig. 5 Our P-Rank difference vs. Baseline difference with iteration number (a) < 7 on P2P; (b) > 6 on P2P; (c) < 11 on Facebook; (d) > 10 on Facebook

superior to the baseline: our approximate PPR difference is lower. And our accurate PPR difference is less than the baseline in some data sets. In the worst case, our accurate PPR difference is equal to the baseline. Also, our P-Rank difference is largely less than the baseline.

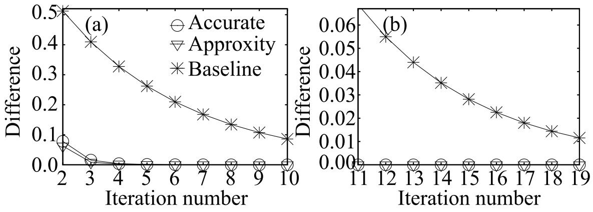

Fig. 6 Our P-Rank difference vs. Baseline difference with iteration number (a) < 11 on Cora; (b) > 10 on Cora

Then we test efficiency of our bounds in the scenario of a top-k node query on two large real data sets (Figs. 7–10). All the top-k queries are repeated 200 times and the reported values are the averages. Our SR bound is 1.5–2.3 times faster than the baseline while it achieves a high precision (⩾ 96%). Our PPR bound is 200–400 times faster than the baseline and it achieves a very high precision (⩾ 97%). Our P-Rank bound is 1–10 times faster than the baseline and it achieves a very high precision (⩾ 98.8%).

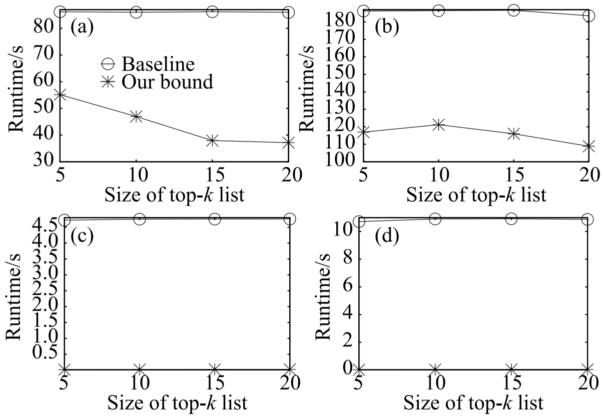

Fig. 7 Runtime of top-k query. (a) SR on Ego-Twitter; (b) SR on Web-Stanford; (c) PPR on Ego-Twitter; (d) PPR on Web-Stanford

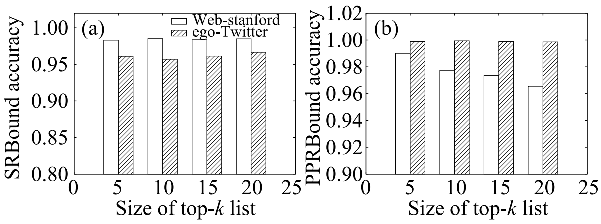

Fig. 8 Accuracy of top-k query. (a) SR bound accuray; (b) PPR bound accuray

Our bounds significantly outperform the state-of-the-art upper bounds.

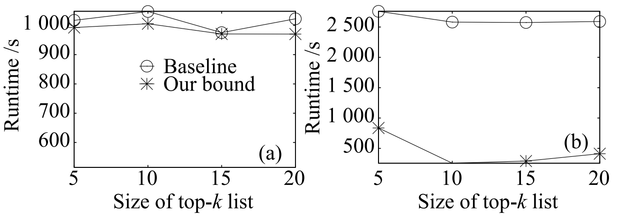

Fig. 9 Runtime of top-k query. (a) P-Rank on Web-Stanford; (b) P-Rank on Ego-Twitter

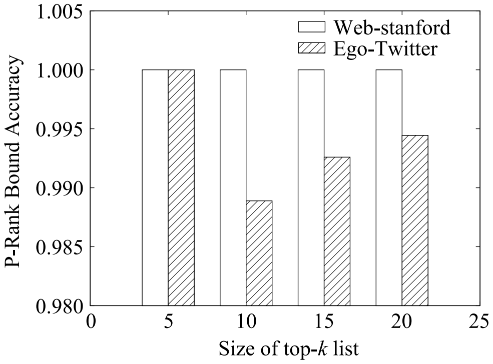

Fig. 10 Accuracy of top-k query

6 Related work

Recently, link-based similarity measures have attracted the attention of many researchers.

Optimization computation The Personalized PageRank (PPR), SimRank (SR) and Penetrating Rank (P-Rank) involve an infinite number of random walks. This naturally incurs heavy overhead of computation.

- SR Lizorkin et al. proposed three excellent optimization methods that improve the time cost from $O(kn^4)$ to $O(knl)$ where k is the number of iterations, n is the number of nodes, and l is the number of edges [14]. They also give a precise accuracy estimate for SR iterative computation, which we discussed in Section 1. Observe that computations of different partial sums may have duplicate redundancy [14]. Therefore, Yu et al. eliminate partial sum redundancy using an adaptive clustering strategy [17]. In their work they also proposed a variant of SR and gave a corresponding difference between theoretical and iterative scores. In contrast, we focus on the upper bound of original SR. Based on Eqs. (6) and (17), Zhang et al presented an optimization algorithm that improves the time cost from $O(kn^4)$ to $O(knl)$ [13]. According to their results, the optimization algorithm outperforms the partial sums method.
- PPR Computing and storing all possible personalized views in advance is impractical [5]. Jeh et al. suggested a scalable solution for PPR based on the observation that PPR

vectors are a linear combination of basis vectors and consider hub-pivoted paths that pass through some important "hub" nodes [5]. Based on Eq. (16), Zhu et al proposed FastPPV, an approximate PPV computation algorithm that is incremental [15]. They proposed L1 error to control accuracy of PPR at query time. The L1 error is defined as follows $\phi^k = 1 - \sum_{p \in V} r_k(q, p)$: L1 error measures overall error, whereas we give PPR an upper bound of any specific node pairs.

- P-Rank Zhao et al introduced a fixed point algorithm for computing P-Rank [9]. Furthermore, they proposed two efficient pruning techniques to reduce space and time complexity: the radius- and category-based pruning techniques. Li et al. proposed an estimation of iterative P-Rank that is defined a general form: the damping factors for in- and out-link directions are different [18]. Our upper bound in Eq. (14) (the base line) is its special case when $c_{out} = c_{in}$. In practice, it is challenging to determine the different values for c_{out} and c_{in}. Therefore, in general the in- and out-link are considered to have the same effect when measuring the similarity. We use the same damping factor as in Ref. [9]: $c_{out} = c_{in} = 0.8$. Yu et al. proposed a probabilistic framework to rapidly compute P-Rank scores [19].

Application of link-based Measure By utilizing upper/lower relevance estimations, the speed of the query, computing top-k relevant nodes w.r.t. a query node, can be accelerated [16, 20].

Considering a general situation where the average in/out-degree is D (D $\geqslant$ 1), Li defined a new average SR/P-Rank upper bound as a function of D [16]. However, networks, such as the Internet, the world wide web, and some social networks, are found to have degree distributions that approximately follow a power law [21]. In other words, these networks are highly right-skewed, meaning that a large majority of nodes have low degree but a small number have high degree. On the other hand, the top-k similarity search method only accesses the local neighborhood of the query node. Therefore, the average SR/P-Rank upper bound does not reflect the true local information.

Fujiwara et al. suggested an approach to find the top-k nodes so as to support interactive similarity search based on PPR [20]. To compute the upper similarity bound, they utilized R_i, the set of nodes that is reachable by any node in S_i for which they would update lower and upper similarity bounds. However, according to the work of Jin et al., the method that tells whether a vertex u can reach another vertex v is time-consuming [22]. In contrast, our upper bound has comparatively low overhead.

Sun et al. proposed a link-based similarity join (LS-join) that extends the similarity join operator to link-based measures [11]. They accelerated the speed of the join query by utilizing an upper bounds of PPR and SR. The upper bounds in [11] are used as a baseline to compare with our upper bounds.

Zheng et al. proposed an estimated shortest-path distance based upper bound for SR [23]. However, it is expensive to compute the shortest path between two vertices on the fly. Furthermore, as with the upper bound in Ref. [10], the shortest-path distance based upper bound is also coarse.

7 Conclusion

We proposed upper bounds of PPR, SR, and P-Rank that are based on the following intuition: the smaller the difference between the two consecutive iteration steps is, the smaller the difference between the theoretical and iterative similarity scores becomes. Our upper bounds are accurate and can easily be achieved. Furthermore, we customize our bounds to accelerate top-k similar nodes query. Our experiments show that our upper bounds significantly outperforms the state-of-the-art upper bounds.

Acknowledgements This work was supported by National Basic Research Program of China (973 Program) (2014CB340402, 2012CB316205), NSFC (Grant Nos. 61272137, 61033010, 61165004), National Social Science Foundation key project (12&ZD220) and National High Technology Research and Development Program of China (863) (2014AA015200). It was partially done when the authors worked in SA Center for Big Data Research in RUC. This Center is funded by a Chinese National 111 Project Attracting.

References

1. Gupta P, Goel A, Lin J, Sharma A, Wang D, Zadeh R. WTF: the who to follow service at Twitter. In: Proceedings of International World Wide Web Conference. 2013, 505–514
2. Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. Journal of the Association for Information Science and Technology, 2007, 58(7): 1019–1031
3. Joshi A, Kumar R, Reed B, Tomkins A. Anchor-based proximity measures. In: Proceedings of International World Wide Web Conference. 2007, 1131–1132
4. Antonellis I, Molina H G, Chang C C. SimRank++: query rewriting through link analysis of the click graph. Proceedings of the VLDB Endowment, 2008, 1(1): 408–421
5. Jeh G, Widom J. Scaling personalized web search. In: Proceedings of International World Wide Web Conference. 2003, 271–279
6. Jeh G, Widom J. SimRank: a measure of structural-context similarity. In: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2002, 538–543

7. Sarkar P, Moore A W, Prakash A. Fast incremental proximity search in large graphs. In: Proceedings of International Conference on Machine Learning. 2008, 896–903
8. Sarkar P, Moore A W. A tractable approach to finding closest truncated-commute-time neighbors in large graphs. In: Proceedings of Uncertainty in Artificial Intelligence. 2007, 335–343
9. Zhao P, Han J, Sun Y. P-Rank: a comprehensive structural similarity measure over information networks. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. 2009, 553–562
10. Lizorkin D, Velikhov P, Grinev M N, Turdakov D. Accuracy estimate and optimization techniques for SimRank computation. Proceedings of the VLDB Endowment, 2008, 1(1): 422–433
11. Sun L, Cheng R, Li X, Cheung D W, Han J. On link-based similarity join. The Proceedings of the VLDB Endowment, 2011, 4(11): 714–725
12. Zhang Y, Li C, Xie C, Chen H. Accuracy estimation of link-based similarity measures and its application. In: Proceedings of Web-Age Information Management WAIM. 2014, 100–112
13. Zhang Y, Li C, Chen H, Sheng L. Fast SimRank computation over disk-resident graphs. In: Proceedings of International Conference of Database Systems for Advanced Applications. 2013, 16–30
14. Lizorkin D, Velikhov P, Grinev M N, Turdakov D. Accuracy estimate and optimization techniques for SimRank computation. The International Journal on Very Large Data Bases, 2010, 19(1): 45–66
15. Zhu F, Fang Y, Chang K C C, Ying J. Incremental and accuracy-aware personalized PageRank through scheduled approximation. The Proceedings of the VLDB Endowment, 2013, 6(6): 481–492
16. Lee P, Lakshmanan L V S, Yu J X. On top-*k* structural similarity search. In: Proceedings of International Conference on Data Engineering. 2012, 774–785
17. Yu W, Lin X, Zhang W. Towards efficient simrank computation on large networks. In: Proceedings of International Conference on Data Engineering. 2013, 601–612
18. Li X, Yu W, Yang B, Le J. ASAP: Towards accurate, stable and accelerative penetrating-rank estimation on large graphs. In: Proceedings of Web-Age Information Management. 2011, 415–429
19. Yu W, Le J, Lin X, Zhang W. On the efficiency of estimating penetrating rank on large graphs. In: Proceedings of Scientific and Statistical Database Management. 2012, 231–249
20. Fujiwara Y, Nakatsuji M, Shiokawa H, Mishima T, Onizuka M. Efficient ad-hoc search for personalized pagerank. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 2013, 445–456
21. Albert R, Barabasi A. Statistical mechanics of complex networks. Reviews of Modern Physics, 2002, 74: 47–97
22. Jin R, Ruan N, Xiang Y, Wang H. Path-tree: an efficient reachability indexing scheme for large directed graphs. ACM Transaction Database System, 2011, 36(1): 1–44
23. Zheng W, Zou L, Feng Y, Chen L, Zhao D. Efficient simrank-based similarity join over large graphs. Proceedings of the VLDB Endowment, 2013, 6(7): 493–504

Yinglong Zhang received his PhD from RenMin University, China in 2014. He is a lecturer at China East Jiaotong University, China. His research interests include data mining and information network analysis.

Cuiping Li received her PhD from the Chinese Academy of Sciences, China in 2003. She is a professor and doctoral supervisor at Renmin University, China. Her research interests include databases, data mining, information network analysis, and data stream management.

Chengwang Xie received his PhD from Wuhan University, China in 2010. He is an associate professor at East China Jiaotong University, China. His research interests include evolutionary computation and data miming.

Hong Chen received her PhD from the Chinese Academy of Sciences, China in 2000. She is a professor and doctoral supervisor at Renmin University, China. Her research interests include databases, data mining, data stream analysis and management, and sensor network data management.