P-Rank: a Comprehensive Structural Similarity Measure over Information Networks

Peixiang Zhao Jiawei Han Yizhou Sun
Department of Computer Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.
pzhao4@uiuc.edu hanj@cs.uiuc.edu sun22@uiuc.edu

ABSTRACT

With the ubiquity of information networks and their broad applications, the issue of similarity computation between entities of an information network arises and draws extensive research interests. However, to effectively and comprehensively measure "*how similar two entities are within an information network*" is nontrivial, and the problem becomes even more challenging when the information network to be examined is massive and diverse. In this paper, we propose a new similarity measure, P-Rank (**P**enetrating **Rank**), toward effectively computing the structural similarities of entities in real information networks. P-Rank enriches the well-known similarity measure, SimRank, by jointly encoding both in- and out-link relationships into structural similarity computation. P-Rank is proven to be a unified structural similarity framework, under which all state-of-the-art similarity measures, including CoCitation, Coupling, Amsler and SimRank, are just its special cases. Based on its recursive nature of P-Rank, we propose a fixed point algorithm to reinforce structural similarity of vertex pairs beyond the localized neighborhood scope toward the entire information network. Our experimental studies demonstrate the power of P-Rank as an effective similarity measure in different information networks. Meanwhile, under the same time/space complexity, P-Rank outperforms SimRank as a comprehensive and more meaningful structural similarity measure, especially in large real information networks.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Graph algorithms; H.2.8 [**Database Applications**]: Data mining

General Terms

Algorithms, Measurement, Performance, Reliability

Keywords

Structural similarity, Information network, Graph mining

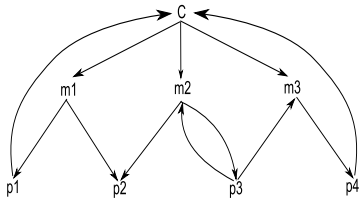
CIKM'09, November 2–6, 2009, Hong Kong, China.

1. INTRODUCTION

Social and technical information systems usually consist of a large number of interacting physical, conceptual, and human/societal entities. Such individual entities are interconnected to form large and sophisticated networks. Without loss of generality, we call these interconnected networks as *information networks* (*IN*s). Examples of INs include the Web [15], highway or urban transportation networks [11], research collaboration and publication networks [8], biological networks [21] and social networks [19]. Clearly, INs are ubiquitous and form a critical component of modern information infrastructure.

In this paper, we focus on the problem of similarity computation on entities of INs. Our study is motivated by recent research and applications on proximity query processing, outlier detection, classification and clustering over different INs, which usually require an effective and trustworthy evaluation of underlying similarity functions among entities. It is desirable to propose a comprehensive similarity measure on INs which can both map human intuition and generalize well under different IN settings. However, it is nontrivial to systematically compute entity similarity in a general and effective fashion, and it becomes especially challenging when the INs to be examined are massive and diverse.

In the mean time, multiple aspects of entities in INs can be exploited for similarity computation, and the choices are usually made domain-specifically. In this paper, we propose a new structural similarity measure, P-Rank (**P**enetrating **Rank**), which solely explores the link structure of the underlying INs for similarity computation. Compared with traditional text contents, the link-based structural information is more homogenous and language independent, which is critical for similarity computation [18]. Concretely, within an IN, we compute P-Rank that says "*two entities are similar if (1) they are referenced by similar entities; and (2) they reference similar entities.*" In comparison with the state-of-the-art structural similarity measure, SimRank [10], which considers the first aforementioned factor only, P-Rank encodes both in- and out-link relationships into computation toward a semantically complete and robust similarity measure. Moreover, similarity beliefs of entity pairs are propagated beyond local neighborhood scope to the entire IN, whose global structure is fully utilized in order to reinforce similarity beliefs of entites in a recursive fashion. P-Rank is also proven to be a general framework for structural similarity of INs and can easily be adapted in any IN settings wherever there exist enough interlinked relationships among entities. For practical applicability, P-Rank can be effectively



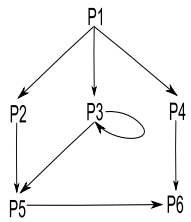
	SimRank	P-Rank
{c, c}	1	1
{m1, m2}	0.4	0.420
{m1, m3}	0.4	0.380
{m2, m3}	0.4	0.295
{p1, p2}	0.56	0.283
{p1, p3}	0.32	0.176
{p3, p4}	0.32	0.124

Figure 1: A Heterogenous IN and Structural Similarity Scores for SimRank **($C = 0.8$) and** P-Rank **($C = 0.8$, $\lambda = 0.5$)**

coupled with other non-structural domain-specific similarity measures, for example, textual similarity, toward a unified similarity measure for INs.

Example 1.1: (A Heterogeneous IN) Consider a heterogeneous information network G in Figure 1, representing a typical submission, bid, review and acceptance procedure of a conference. G is regarded as *heterogeneous* if vertices (entities) of G belong to different mutual exclusive categories, such as $Conference = \{c\}$, $CommitterMember = \{m_1, m_2, m_3\}$ and $Paper = \{p_1, p_2, p_3, p_4\}$. Directed edges model the relationships between vertices in different categories. Two structural similarity measures, SimRank and P-Rank, for different vertex pairs of G are illustrated as well. As shown in Figure 1, the conference c is considered similar to itself, and the similarity scores (for both SimRank and P-Rank) are set to be 1. For committee member pairs $\{m_1, m_2\}$, $\{m_1, m_3\}$, and $\{m_2, m_3\}$, as both vertices of each pair are pointed to by c (they both are invited as committee members by the conference, c), we may infer that they are similar. However, SimRank cannot differentiate among these three pairs. (They have the same SimRank score, 0.4). The main reason is that for committee member pairs, SimRank considers the in-link relationships with the vertex c only, while neglecting out-link relationships with paper vertices $\{p_1, p_2, p_3, p_4\}$. P-Rank, however, takes into account of both in- and out-link relationships for similarity computation. As to $\{m_1, m_2\}$, for example, because they both point to p_2 (both m_1 and m_2 bid for paper p_2), the structural similarity between them is further strengthened (P-Rank score is 0.420, which is different from that of $\{m_2, m_3\}$ (0.295), and that of $\{m_1, m_3\}$ (0.380)). We generalize this idea by observing that once we have concluded similarity between m_1 and m_2, p_1 and p_3 are similar as well because they are pointed to by m_1 and m_2, respectively, although this inference is somehow weakened during similarity propagation. Continuing forth, for every comparable pair of vertices in G, we can infer P-Rank between them. □

Example 1.2: (A Homogeneous IN) Consider a homogeneous information network G in Figure 2, representing a tiny literature graph. G is *homogeneous* if vertices of G, which represent scientific publications in this example, all belong to one category ("Publication"). Edges between vertices are references/citations from one paper to another. Different from heterogeneous INs, any pair of vertices in homogenous INs can be measured by their structural similarity because they all belong to the same category. We present SimRank and P-Rank scores for some of them, as shown in Figure 2. SimRank cannot tell the differences between the vertex pair $\{P2, P3\}$ and $\{P3, P4\}$, solely because SimRank considers partial relationship information for similarity



	SimRank	P-Rank
{P1, P1}	1	1
{P2, P3}	0.4	0.441
{P3, P4}	0.4	0.364
{P4, P5}	N/A	0.188
{P2, P5}	N/A	0.062

Figure 2: A Homogeneous IN and Structural Similarity Measures for SimRank **($C = 0.8$) and** P-Rank **($C = 0.8$, $\lambda = 0.8$)**

computation. More severely, SimRank is unavailable for the vertex pairs $\{P4, P5\}$ and $\{P2, P5\}$, mainly because these vertex pairs do not have common in-link similarity flows. However, P-Rank can successfully infer structural similarity for all vertex pairs by considering both in- and out-link relationships into computation, thus outperforms SimRank in homogeneous INs. □

As its name dictates, P-Rank encodes both in- and out-link relationships of entities in similarity computation, *i.e.*, P-Rank scores flow from in-link neighbors of entities and penetrate through their out-link ones. Furthermore, this process is recursively propagated beyond the localized neighborhood scope of entities to the entire IN. The major merits of P-Rank are its semantic completeness, generality and robustness. As a comprehensive structural similarity measure, P-Rank can be effectively adapted in INs with different variety and scale, in which most up-to-date similarity measures, like SimRank, may generate biased answers or simply fail due to the incomplete structural information considered during similarity computation, as illustrated in Example 1.1 and Example 1.2. In order to compute P-Rank efficiently, we propose an iterative algorithm which converges fast to a fixed-point. The correctness of the algorithm is proven that the iterative algorithm always converges to its theoretical upper bound.

The contributions of this paper are summarized as follows:

1. We propose a new structural similarity measure, P-Rank, which is applicable in INs. We study its mathematical properties, its advantages over other state-of-the-art structural similarity measures, and its derivatives under different IN settings.

2. We propose a fixed-point iterative P-Rank algorithm toward effectively computing P-Rank in INs. We prove the correctness of the algorithm and discuss the optimization techniques to facilitate P-Rank computation in different scenarios.

3. P-Rank is shown to be a unified structural similarity framework in INs, under which the well-known structural similarity measures, CoCitation, Coupling, Amsler and SimRank , are all its special cases.

4. We do extensive experimental studies on both real and synthetic data sets. The evaluation results demonstrate the power of P-Rank as a general structural similarity measure for different INs.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, we present our structural similarity measure, P-Rank, from both mathematical

and algorithmic perspectives. We report our experimental studies in Section 4. Section 5 concludes the paper.

2. RELATED WORK

As common standards to determine the closeness of different objects, similarity (or proximity) measures are crucial and frequently applied in clustering, nearest neighbor classification, anomaly detection and similarity query processing. Compared with traditional textual contents, link-based structural context in INs is of special importance and exploited frequently in similarity computation. In previous studies, SimFusion [23] aimed at "combining relationships from multiple heterogeneous data sources". [16] proposed a similarity measure based on PageRank score propagation through link paths. [7] explored methods for ranking partial tuples in a database graph. Maguitman *et al.* did extensive comparative studies on different similarity measures [18], and the results demonstrate that link-based structural similarity measures produce systematically better correlation with human judgements compared to the text-based ones.

In bibliometrics, similarities between scientific publications are commonly inferred from their cross-citations. Most noteworthy from this field are the methods of CoCitation [22], Coupling [13] and Amsler [1]. For CoCitation, the similarity between two papers p and q is based on the number of papers which reference both p and q. For Coupling, the similarity is based on the number of papers referenced by both p and q. Amsler fuses both CoCitation and Coupling for similarity computation. These methods have been efficiently applied to cluster scientific publications and web pages [20].

SimRank [10, 6, 17], together with its variant [2], is an iterative PageRank-like structural similarity measure for INs. It goes beyond simple CoCitation much as PageRank goes beyond direct linking for computing importance of web pages. The weakness of SimRank, called the ***limited information problem***, is discussed in [10]. SimRank makes use of in-link relationships only for similarity computation while neglecting similarity beliefs conveyed from out-link directions. Therefore, the structural information of INs is partially exploited and the similarity computed is inevitably asymmetric and biased. In real INs, those "*unpopular entities*", *i.e.*, entities with very few in-link relationships will be penalized by SimRank. More severely, SimRank can even be unavailable for entities with no in-link similarity flows (shown in Example 1.2). However, those entities with few or no in-links are dominating the INs in quantity, as expressed by the power law distribution and heavy-tailed in(out)-degree distribution [4]. Meanwhile, these entities are often not neglectable because they are new, potentially popular, and interesting to most users. However, they tend to be harder for humans to find. To overcome the limited information problem of SimRank, we propose P-Rank which refines structural similarity by jointly considering both in- and out-link relationships of entity pairs. Furthermore, the similarity computation goes beyond the localized neighborhood so that the global structural information of INs are exploited to reinforce similarity beliefs of entities. As discussed in the reminder of the paper, with the same time/space complexity as SimRank, P-Rank can achieve much better results and solve the limited information problem effectively. Heymans *et al.* [9] proposed similar ideas to model structural similarity of enzymes in metabolic pathway graphs in order for the phylogenetic analysis of metabolic pathways. However, their similarity are on vertices in different graphs and if the factors of dissimilarity and absence of edges are not considered, their work can be regarded as a special case of P-Rank ($C = 1$ and $\lambda = 0.5$).

Iterative fixed-point algorithms over the web graph, like HITS [14] and PageRank [3], have been studied and applied to compute "importance" scores for Web pages. Results show that the usage of structure of INs can greatly improve search performance versus text alone.

3. P-RANK

The basic recursive intuition of P-Rank can be expressed as "*two entities in an IN are similar if they are related to similar entities*". More specifically, the two-fold meaning of P-Rank is elaborated as

1. *two entities are similar if they are referenced by similar entities*
2. *two entities are similar if they reference similar entities*

As the base case, we consider an entity maximally similar to itself, to which we can assign the P-Rank score of 1. (If other entities are known to be similar *a-priori*, their similarities can be pre-assigned as well.) For each pair of distinct entities, we take into consideration both their in- and out-link relationships for similarity computation. This similarity is then penetrating from in-link neighbors to out-link ones and propagated toward the entire IN.

3.1 Preliminaries

We model an IN as a labeled directed graph $G = (V, E, \Sigma; l)$ where vertex $v \in V$ represents an entity of the domain and a directed edge $\langle u, v\rangle \in E$ represents a relationship from entity u to entity v, where $u, v \in V$. Σ is an alphabet set and $l : V \rightarrow \Sigma$ is a labeling function. In heterogeneous INs, $V = \{V_1 \bigcup V_2 \cdots \bigcup V_n\}$ can be partitioned into n mutual exclusive vertex subsets, $V_1, V_2, \cdots, V_n$, $V_i \bigcap V_j = \emptyset$ for $1 \leq i, j \leq n$, which belong to n different domain-specific categories. In homogeneous INs, however, there is no distinction among vertices. Note that our definition of INs can be naturally extended to undirect graph or edge-weighted graph settings.

For a vertex v in a graph G, we denote by $I(v)$ and $O(v)$ the set of in-link neighbors and out-link neighbors of v, respectively. Note that either $I(v)$ and $O(v)$ can be empty. An individual in-link neighbor is denoted as $I_i(v)$, for $1 \leq i \leq |I(v)|$, if $I(v) \neq \emptyset$, and an individual out-link neighbor is denoted as $O_i(v)$, for $1 \leq i \leq |O(v)|$, if $O(v) \neq \emptyset$.

3.2 P-Rank Formula

We denote the P-Rank score for vertex a and b by $s(a, b) \in [0, 1]$. Following our aforementioned intuition, P-Rank can be formalized recursively in Equation (1), when $a \neq b$:

$$\begin{aligned} s(a,b) = \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \\ + (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} s(O_i(a), O_j(b)) \end{aligned} \tag{1}$$

Otherwise, P-Rank is defined as

$$s(a, b) = 1 \tag{2}$$

In Equation (1), the relative weight of in- and out-link directions is balanced by parameter $\lambda \in [0,1]$. C is set as a damping factor for in- and out-link directions, $C \in [0,1]$ [1]. The reason is that $s(a,b)$ will be attenuated during similarity propagation. When $I(a)$(or $I(b)$) $= \emptyset$, the in-link part is invalidated and only the out-link direction takes into effect. Similarly, when $O(a)$(or $O(b)$) $= \emptyset$, only the similarity flows from in-link part are considered. If both $I(a)$(or $I(b)$) $= \emptyset$ and $O(a)$(or $O(b)$) $= \emptyset$, we define $s(a,b) = 0$.

Equation (1) is written for every pair of vertices $a, b \in G$, resulting in a set of n^2 equations for a graph of size n. To solve the set of n^2 equations, we rewrite the recursive P-Rank formula (shown in Equation (1)) into the following iterative form

$$R_0(a,b) = \begin{cases} 0 & (\text{if } a \neq b) \\ 1 & (\text{if } a = b) \end{cases} \tag{3}$$

and

$$\begin{aligned} R_{k+1}(a,b) = \ & \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b)) \\ & + (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} R_k(O_i(a), O_j(b)) \end{aligned} \tag{4}$$

where $R_k(a,b)$ denotes the P-Rank score between a and b on iteration k, for $a \neq b$ and $R_k(a,b) = 1$ for $a = b$. We progressively compute $R_{k+1}(*,*)$ based on $R_k(*,*)$. That is, on iteration $(k+1)$, we update $R_{k+1}(a,b)$ by the P-Rank scores from the previous iteration k. This iterative computation starts with $R_0(*,*)$ where $R_0(a,b)$ is a lower bound of the actual P-Rank score, $s(a,b)$.

Theorem 3.1: The iterative P-Rank equations (shown in Equation (3) and Equation (4)) have the following properties

1. (***Symmetry***) $R_k(a,b) = R_k(b,a)$
2. (***Monotonicity***) $0 \leq R_k(a,b) \leq R_{k+1}(a,b) \leq 1$
3. (***Existence***) The solution to the iterative P-Rank equations always exists and converges to a fixed point, $s(*,*)$, which is the theoretical solution to the recursive P-Rank equations.
4. (***Uniqueness***) the solution to the iterative P-Rank equations is unique when $C \neq 1$.

Proof: Shown in Appendix. □

Theorem 3.1 demonstrates four important properties of P-Rank. For any vertices $a, b \in G$, the iterative P-Rank between a and b is the same as that between b and a, *i.e.*, P-Rank is a symmetric measure, as mentioned in property 1 (*Symmetry*). Property 2 (*Monotonicity*) shows that the iterative P-Rank is non-decreasing during similarity computation. However, the solution will not go to infinity. Property 3 (*Existence*) and 4 (*Uniqueness*) guarantee that there exists a unique solution to n^2 iterative P-Rank equations, which can be reached by iterative computation to a fixed point, *i.e.*, the solution to iterative P-Rank converges to a limit which satisfies the recursive P-Rank equation, shown in Equation (1):

$$\forall\ a, b \in G, \lim_{k \to \infty} R_k(a,b) = s(a,b) \tag{5}$$

In real applications, iterative P-Rank converges very fast (details are shown in Section 4). Empirically, we can choose to fix a small number of iterations ($k \approx 5$) to derive P-Rank for all pair of vertices in real INs.

[1]For a more general form of P-Rank, C can be replaced by two different parameters C_{in} and C_{out} to represent damping factors for in- and out-link directions, respectively. We omit the details as it is fairly easy to extend our work into that scenario.

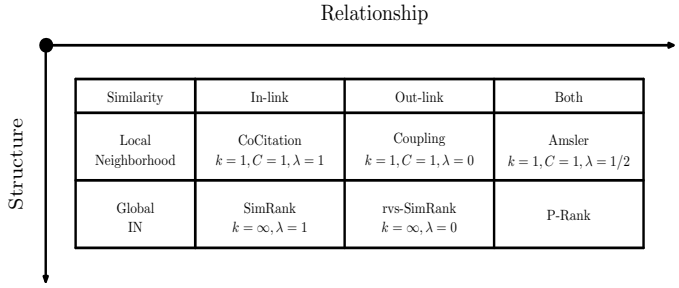


Figure 3: Structural Similarity Matrix for INs

3.3 Derivatives of P-Rank

Besides its semantic completeness with a consideration of both in- and out-link relationships in similarity computation, P-Rank outperforms other structural similarity measures by its generality and flexibility. As shown in Figure 3, all of the state-of-the-art structural similarity measures proposed so far for INs are illustrated in the structural similarity matrix. Among all measures shown in Figure 3, P-Rank enjoys the most general form, from both the semantic completeness perspective and the structure perspective. All other measures, such as CoCitation, Coupling, Amsler and SimRank, are just simplified special cases of P-Rank and can be easily derived from P-Rank. P-Rank therefore provides a unified framework for structural similarity computation in INs. By analyzing the iterative P-Rank shown in Equation (4), we can draw the following conclusions:

1. When $k = 1$, $C = 1$ and $\lambda = 1$, P-Rank is reduced to CoCitation.
2. When $k = 1$, $C = 1$ and $\lambda = 0$, P-Rank is reduced to Coupling.
3. When $k = 1$, $C = 1$ and $\lambda = 1/2$, P-Rank is reduced to Amsler, which subsumes both CoCitation and Coupling. Amsler can be regarded as a *one-step* P-Rank without similarity propagation.
4. When $k \to \infty$ and $\lambda = 1$, P-Rank boils down to SimRank, which is an iterative form of CoCitation with no out-link similarity considered.
5. When $k \to \infty$ and $\lambda = 0$, P-Rank is degenerated to a new structural similarity measure, which is an iterative form of Coupling with no in-link similarity involved. Since this new measure considers out-link relationships only and is the counterpart of SimRank, we name it rvs-SimRank, short for *reverse*-SimRank. In real INs, rvs-SimRank is more practical and useful than SimRank, because entities of a massive IN are usually highly distributed. It is prohibitive to maintain a global view of

the whole IN. And an entity usually has a good knowledge of what entities are referenced by it, but it is hard to know what entities are referencing it without a full scan of the entire IN. For example, a web page contains hyper-links to other web pages for its own sake, but it is impossible to know which web pages are hyper-linking it without examining the whole Web beforehand. This becomes even more severe when INs are dynamically changing over time. So, rvs-SimRank is more robust and adaptive for measuring structural similarity over large yet dynamically changed INs.

In real applications, P-Rank can be adapted flexibly to different IN settings, as long as there exist enough interlinked relationships between entities. Even when the IN to be studied has sparse in-link information or biased edge distribution in which SimRank may fail, P-Rank can still work well in modeling structural similarities.

Another important issue is to select appropriate values for parameters C, λ and k in P-Rank computation. C represents the degree of attenuation in similarity propagation, and λ expresses the relative weight of similarity computation between in-link and out-link directions. A priori knowledge of the IN infrastructure is usually helpful to select the values of C and λ. By sampling a set of subgraphs from the original IN, we can learn the characteristics of the underlying IN, and C, λ can be set based on the sampled subgraphs as an approximation. The convergence of iterative P-Rank is fast with only several iterations of computation, so k is usually set empirically as a small constant number. In Section 4, we will systematically study the effects of different parameters on P-Rank computation.

3.4 Computing P-Rank

Based on Section 3.2, the solution to the recursive P-Rank formula (Equation (1)) can be reached by computing its iterative form (Equation (4)) to a fixed point. Algorithm 1 illustrates the iterative procedure for computing P-Rank in an IN, G. Let n be the number of vertices in G and k be the number of iterations executed until P-Rank converges to its fixed point. For every vertex pair (a, b), an entry $R(a, b)$ maintains the intermediate P-Rank score of (a, b) during iterative computation. Because the $(k+1)$-th iterative P-Rank score is computed based on P-Rank scores in the k-th iteration, an auxiliary data structure $R^*(a, b)$ is maintained accordingly. As proven in Theorem 3.1(1), $R_k(a, b) = R_k(b, a)$, so only one order for each pair is stored explicitly. In real implementations, either sparse matrixes or hash tables can be chosen as core data structures for $R(\cdot, \cdot)$ and $R^*(\cdot, \cdot)$. Because G can be so large as not to be held in main memory, any advanced data structures that optimize external memory accesses can be applied.

Algorithm 1 first initializes $R_0(a, b)$ based on Equation 3 (Lines $1 - 4$). During iterative computation, P-Rank in $(k+1)$-th iteration, $R^*(\cdot, \cdot)$, is updated by $R(\cdot, \cdot)$ in the k-th iteration, based on Equation 4 (Lines $6 - 18$). Then $R(\cdot, \cdot)$ is substituted by $R^*(\cdot, \cdot)$ for further iteration (Lines $19 - 21$). This iterative procedure stabilizes rapidly and converges to a fixed point within a small number of iterations. A typical call to the algorithm can be P-Rank (G, 0.5, 0.8, $\lceil ln(n) \rceil$), where the relative weight λ is set to be 0.5 and the damping factor C is set to be 0.8.

The space complexity of Algorithm 1 is $O(n^2)$, the amount to store intermediate and final P-Rank scores of G, *i.e.*, the size of $R^*(\cdot, \cdot)$ and $R(\cdot, \cdot)$. Let d_1 and d_2 be the average in-degree and out-degree over all vertices of G, respectively, the time complexity of the algorithm is $O(k(d_1^2 + d_2^2)n^2)$, and the worst case time complexity can be $O(n^4)$. In comparison with SimRank whose space and time complexities are $O(n^2)$ and $O(n^4)$, P-Rank has the same space and time complexities with SimRank.

Algorithm 1: P-Rank (G, λ, C, k)

Input : An IN G, the relative weight λ, the damping factor C, the iteration number k
Output: P-Rank score $s(a, b)$, $\forall\, a, b \in G$

```
1  foreach a ∈ G do                          /* Initialization */
2      foreach b ∈ G do
3          if a == b then R(a,b) = 1
4          else R(a,b) = 0
5  while (k > 0) do                          /* Iteration */
6      k ⟵ k − 1
7      foreach a ∈ G do
8          foreach b ∈ G do
9              in ⟵ 0
10             foreach i_a ∈ I(a) do
11                 foreach i_b ∈ I(b) do
12                     in ⟵ in + R(i_a, i_b)
13             R*(a,b) ⟵ λ * C*in / (|I(a)||I(b)|)
14             out ⟵ 0
15             foreach o_a ∈ O(a) do
16                 foreach o_b ∈ O(b) do
17                     out ⟵ out + R(o_a, o_b)
18             R*(a,b) += (1 − λ) * C*out / (|O(a)||O(b)|)
19     foreach a ∈ G do                      /* Update */
20         foreach b ∈ G do
21             R(a,b) = R*(a,b)
22 return R(*,*)
```

In [17], the authors improved the time complexity of SimRank from $O(n^4)$ to $O(n^3)$. The same memoization based algorithms can be applied in the same way on P-Rank to reduce its time complexity to $O(n^3)$. In [6], the authors suggested a scalable framework for SimRank computation based on the Monte Carlo method. Essentially their computation is probabilistic and the SimRank scores computed are an approximation to the exact answer. In order to make full use of characteristics of different INs, we propose different pruning algorithms to efficiently compute P-Rank.

Homogeneous Information Network: In homogeneous INs, all vertices of G are of the same type. One way to reduce the space/time complexities in this scenario is to prune less similar vertex pairs while not deteriorating the accuracy of similarity computation too much. For n^2 vertex-pair of G, only those adjacent to each other (say, vertices within a radius of 3 or 4) are similar, while those whose neighborhood have little or no overlap are far apart and inevitably not similar. Thus *radius-based pruning* [10] can be used to set the similarity between two vertices far apart to be 0, and only those vertex-pairs within a radius of r from each other in the underlying undirected graph G' are considered in similarity computation. Given a vertex $u \in G$, let there be d_r such neighbors of u within a radius r on the underlying undirect graph G' on average, then there will be $(n * d_r)$

vertex-pairs considered. The space and time complexities become $O(n * d_r)$ and $O(k(d_1^2 + d_2^2)d_r n)$, respectively. Since d_r is likely to be much less than n, if r is small *w.r.t.* n, we can think of this approximate algorithm as being linear with a possibly large constant factor.

Heterogeneous Information Network: In heterogeneous INs, vertices of G belong to different categories. Given two vertices $u, v \in G$, it is meaningless to measure structural similarity between u and v if they belong to different categories. Thus the pruning technique in this scenario, called *category-based pruning*, is to set the similarity between two vertices belonging to different categories to be 0, and consider only those vertex pairs within the same category. Let there be c different categories over the vertices of G, and for each category i, there be n_i vertices included, where $1 \leq i \leq c$, then the total number of vertex pairs is $\sum_{i=1}^{c} n_i^2$. The space and time complexities then become $O(\sum_{i=1}^{c} n_i^2)$ and $O(k(d_1^2+d_2^2)(\sum_{i=1}^{c} n_i^2))$. Notice the following inequality holds,

$$n^2 = (\sum_{i=1}^{c} n_i)^2 \geq \sum_{i=1}^{c} n_i^2$$

Category-based pruning can eliminate a huge number of vertex pairs belonging to different categories, especially when c is large. If the number of vertices in a specific category is still so large that they cannot be held in main memory, radius-based pruning can be further applied within this category to facilitate the computation. [24] presented an advanced index-based algorithm, SimTree, for fast computation of similarity scores in heterogeneous INs if vertices in every category are hierarchically organized. Our category-based pruning algorithm is actually the specialized one-level SimTree.

4. EXPERIMENT

In this section, we report our experimental studies on the effectiveness of P-Rank as a comprehensive structural similarity measure over different INs. We show the power of P-Rank in comparison with the state-of-the-art structural similarity measure, SimRank. In addition, the experiments illustrate the feasibility and efficiency of the P-Rank algorithm with pruning techniques in INs with different diversity and scale.

We ran our experiments on two different datasets: one is real data from DBLP[2] and the other is synthetic [5]. For the real dataset, we further generate two different INs: one is a heterogeneous IN and the other is a homogeneous IN. All our experiments are performed on an Intel PC with a 2.4GHz CPU, 2GB memory, running Redhat Fedora Core 4. All algorithms including P-Rank and SimRank are implemented in C++ and compiled by gcc 3.2.3. For ease and fairness of comparison, we set the damping factor $C = 0.8$ for both SimRank and P-Rank; The relative weight λ is set to be 0.5 for P-Rank , if not specified explicitly. All the default values of parameters are set in accordance with [10].

4.1 A Heterogenous IN from DBLP

We first build a heterogeneous INs from DBLP. The downloaded DBLP data had its time stamp on March 15th, 2008. The heterogeneous IN, G, contains four different kinds of

[2]http://www.informatik.uni-trier.de/ ley/db/

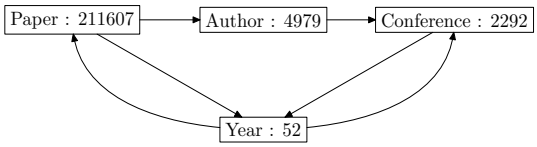


Figure 4: The Schema of the Heterogeneous IN from DBLP Datasets. (The Number within Each Rectangle Represents the Number of Vertices in the Corresponding Category.)

vertices: *paper*, *author*, *conference* and *year*. If a paper p is written by an author a, there exists a directed edge from p to a; If an author a participated in a conference c, there exists a directed edge from a to c; For a specific year y, there are bidirectional edges between both p and y and c and y, if the paper p was published in conference c in year y. Figure 4 illustrates the global schema of the heterogenous IN, G. The number of vertices in G is 218930 and the number of edges is 818301. More specifically, the number of paper vertices is 211607; the number of author vertices is 4979; the number of conference vertices is 2292 and the number of year vertices is 52.

In order to evaluate the effectiveness of P-Rank, we choose to test how different structural similarity measures perform in clustering authors in G. It is worth noting that P-Rank is not confined only in clustering applications. Any data management applications adopting structural similarity as an underlying function can make use of P-Rank as its similarity measure. Meanwhile, P-Rank is orthogonal to the specific clustering algorithms applied, *i.e.*, P-Rank proposes a general structural similarity measure which can be applied in most existing clustering algorithms. We plug P-Rank and SimRank into K-Medoids [12], respectively. The structural distance between two vertices $u, v \in G$ is defined as

$$d_f(u, v) = 1 - s_f(u, v) \tag{6}$$

where $s_f(u, v)$ is the similarity score generated by the similarity function, f, (either p for P-Rank or s for SimRank). We define *compactness* of the clustering results, C_f, as

$$C_f = \frac{\sum_{i=1}^{K} \sum_{x \in C_i} d(x, m_i)}{\sum_{1 \leq i < j \leq K} d(m_i, m_j)} \tag{7}$$

where K is the number of clusters to be generated[3]; C_i is the i-th cluster; m_i, m_j are centers for cluster i and cluster j, respectively. Intuitively, the numerator of Equation (7) describes *intra-cluster* distances and the denominator represents *inter-cluster* distances. Smaller C_f values demonstrate better clustering performance. In the following experiments, we compare C_p and C_s for P-Rank and SimRank, respectively.

We run both P-Rank and SimRank over G until the similarity scores converge. We then cluster author vertices by K-Medoids algorithm, and $K = 10$. At the beginning, we randomly choose 10 author vertices (without replacement) as initial centers of clusters and run the K-Medoids algorithm. We perform $l = 10$ trials and the clustering results are shown in Figure 6. As illustrated, P-Rank consistently achieves more compact clustering results than does SimRank. The main reasons are as follows: (1) P-Rank considers similarity

[3]Note K is different from k in Equation (4), which is the number of iterations performed for iterative P-Rank.

1	Irith Pomeranz, Sudhakar M. Reddy
2	Pankaj K. Agarwal, Micha Sharir
3	Robert K. Brayton, Alberto L. Sangiovanni-Vincentelli
4	Amr El Abbadi, Divyakant Agrawal
5	Didier Dubois, Henri Prade
6	Wei-Ying Ma, HongJiang Zhang
7	Oscar H. Ibarra, Tao Jiang
8	Jiawei Han, Philip S. Yu
9	Hector Garcia-Molina, Jeffrey D. Ullman
10	Mary Jane Irwin, Mahmut T. Kandemir

(a) **Top-**10 **Similar Author Pairs**

1	Jiawei Han
2	Ming-Syan Chen
3	Charu C. Aggarwal
4	Haixun Wang
5	Kun-lung Wu
6	Joel L. Wolf
7	Wei Fan
8	Daniel M. Dias
9	Wei Wang
10	Jiong Yang

(b) **Top-**10 **Nearest Neighbors of "Philip S. Yu"**

1	Joseph M. Hellerstein
2	Jim Gray
3	Stanley B. Zdonik
4	Michael J. Carey
5	Ugur Cetintemel
6	Philip A. Bernstein
7	Mitch Cherniack
8	David J. DeWitt
9	David Maier
10	Lawrence A. Rowe

(c) **Top-**10 **Nearest Neighbors of "Michael Stonebraker"**

Figure 5: Top-10 **Ranking Results for Author Vertices in DBLP by P-Rank**

propagation from both in-link (paper vertices) and out-link directions (conference vertices), which generates more comprehensive results than does SimRank for clustering authors; (2) By simply considering in-link propagation only, SimRank fails to measure quite a few vertex pairs in G. For SimRank, only those authors who cooperate (either directly or indirectly) on some papers have significant similarity scores, while others are regarded as dissimilar. In comparison, P-Rank is more robust than SimRank. For two author vertices, although they may not cooperate with each other (no in-link propagation), as long as they participate in common conferences (there exists out-link propagation), they are regarded as similar to some extent. Therefore, quite a few vertices which are dissimilar under SimRank's scheme are now similar in P-Rank, which improves the compactness of clustering results.

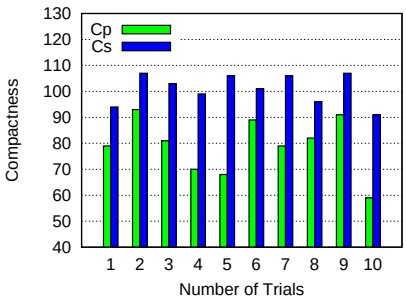
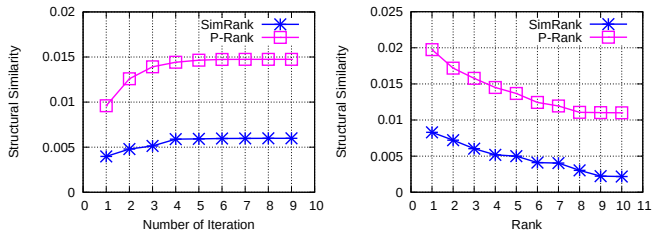


Figure 6: Compactness vs. Number of Trials for P-Rank and SimRank in the Heterogeneous IN

We then test the algorithmic nature and mathematical property of P-Rank. Figure 7(a) plots structural similarity scores of author pairs *w.r.t.* the number of iterations performed in corresponding algorithms. The scores are averaged by the top 10 highest ranked scores of author pairs for P-Rank and SimRank, respectively. We see from the figure that the intermediate similarity scores $R_k(*,*)$ become more accurate on successive iterations. Iteration 2, which computes $R_2(*,*)$ from $R_1(*,*)$, can be thought of as the first iteration taking advantage of the recursive power of algorithms for similarity computation. Subsequent changes become increasingly minor, suggesting a rapid convergence. The figure also manifests that the fixed point iteration process stabilizes very fast, as the number of iterations, k, is greater than 5. Figure 7(b) plots the structural similarity scores of P-Rank and SimRank *w.r.t.* the rank number, N. The downward curves for both P-Rank and SimRank present a decrease in structural similarity as N increases, which is expected because highly ranked authors are more similar.



(a) Varying # Iteration (b) Varying # Ranking

Figure 7: Similarity Measures on Author Pairs in the Heterogeneous IN from DBLP

We further examine the ground truth generated by P-Rank on author vertices of G to test if it really reflects the reality to single out similar authors from the DBLP dataset. Although the judgement of similarity might be quite subjective and difficult even for human beings, we still find very interesting results by making use of P-Rank. As illustrated in Figure 5(a), the top-10 highly ranked author pairs are listed. We may notice that the author pairs with high P-Rank scores share some common characteristics. First, they are usually co-authors or share quite a few authorities as co-authors. And they are purely dedicated in specific research fields. In the mean time, highly ranked authors are inclined to be clustered into a close related community, in which their authorities are further reinforced. That is also another reason why P-Rank outperforms SimRank in entity clustering, as illustrated in the aforementioned experiment. We further issue k-Nearest Neighbor (KNN) queries to retrieve top-k most similar authors in IN G, given an author vertex q as a query. Figure 5(b) shows the ranked results for the query "Philip S. Yu" and Figure 5(c) shows the ranked results for the query "Michael Stonebraker", where $k = 10$. As illustrated, both results are quite intuitive and conform to our basic understandings. Therefore, P-Rank can be effectively used as an underlying metric for measuring structural similarity in heterogenous INs, and its results obey our common sense pretty well.

4.2 A Homogenous IN from DBLP

After the study of P-Rank on heterogeneous INs, we continue generating a homogeneous IN, G, on the DBLP dataset. The vertex set of G is composed of a subset of papers in DBLP and a directed edge exists from paper u to paper v if u cited v. The number of vertices in the homogenous IN G is 21740, and the number of edges is 65186.

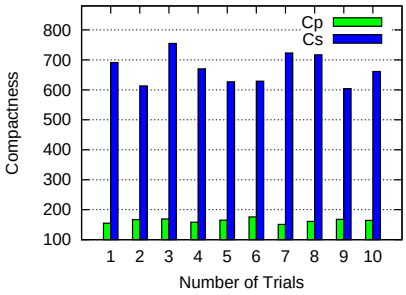


Figure 8: Compactness vs. Number of Trials for P-Rank and SimRank in the Homogeneous IN

Our first experiment is to study how the different structural similarity measures perform in clustering vertices in homogenous IN. We plug P-Rank and SimRank respectively as underlying similarity functions into K-Medoids ($K = 10$). We randomly choose 10 vertices (without replacement) as initial centers of clusters and run the K-Medoids algorithm. We perform $l = 10$ trials and the clustering results are shown in Figure 8. As illustrated, P-Rank can achieve much better results in clustering vertices in homogeneous IN, G. The improvement can be at least 6 times better. And the clustering performance of P-Rank is consistently stable in different experimental trials.

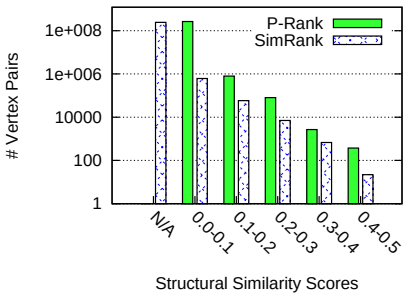


Figure 9: Vertex Pair Distributions Based on P-Rank and SimRank Scores in Homogeneous DBLP IN

Different from heterogenous INs, homogeneous INs have their vertices in one unique category and every vertex pair is eligible for comparison under the P-Rank framework. However, SimRank may fail in homogeneous INs simply because there might be no common in-link similarity flows for vertex pairs. The problem becomes even more severe when the IN is massive and the interlinked relationships are not evenly distributed within the IN. As illustrated in Figure 9, vertex pairs are reorganized into different histograms based on their structural similarity scores computed by P-Rank and SimRank, respectively. For example, vertex pairs whose structural similarity scores are between $[0.1, 0.2)$ are put in the third histogram. A special histogram "N/A" represents vertex pairs whose structural similarity can not be measured properly. Because of the very biased information considered during similarity computation, SimRank fails to generate meaningful similarity measures for a majority of vertex pairs in the homogenous IN, as shown in the histogram "N/A". However, in the real homogenous IN with very uneven relationship distributions, P-Rank can still work well and is robust enough in structural similarity computation.

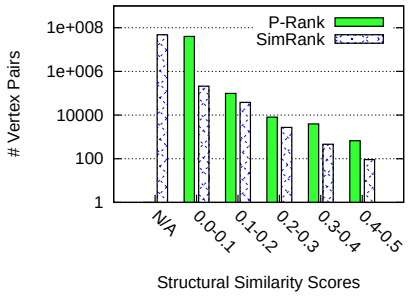
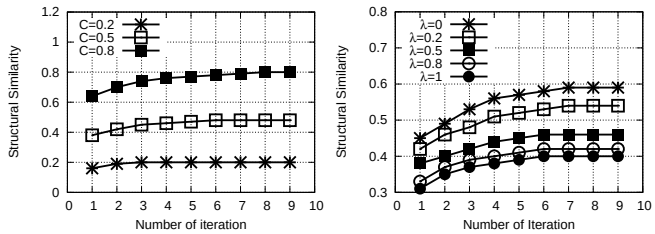


Figure 10: Vertex Pair Distributions Based on P-Rank and SimRank Scores in Homogeneous Synthetic IN

4.3 Synthetic Datasets: R-MAT

We generate a synthetic homogeneous IN G based on the *Recursive Matrix* (R-MAT) model [5], which naturally follows power-law (in- and out-)degree distributions for G. The homogeneous IN G generated is a directed graph with 10^5 vertices and $6 * 10^5$ edges.

In this homogenous IN, we first test how P-Rank and SimRank perform when measuring structural similarity of vertices in G. As illustrated in Figure 10, vertex pairs are distributed to different histograms with different similarity score intervals. Similar to Figure 9, SimRank again fails to deliver meaningful structural similarity for a majority of vertex pairs in IN, as shown in the histogram "N/A". However, P-Rank can successfully measure structural similarity for every pair of vertices in the homogeneous IN, G.



(a) Varying C (b) Varying λ

Figure 11: P-Rank v.s. Different Parameters

We are also interested in how different parameters affect P-Rank when computing similarity in the homogeneous IN. We first test how the damping factor C is correlated with P-Rank. Figure 11(a) illustrates P-Rank scores in G *w.r.t.* the number of iterations performed. The structural similarity scores are averaged by the top 10 highest ranked scores of vertex pairs. The damping factor, C, is set to be 0.2, 0.5 and 0.8, and three curves are plotted, respectively. It is obvious that P-Rank grows proportionally with the increase of C. When $C = 0.2$, P-Rank converges fast when the number of iterations, k, is larger than 2. However, when $C = 0.8$, P-Rank converges approximately at the 7th iteration of computation. The reason is that when C is set to a small value, the recursive power of P-Rank will be weakened and only vertices nearby can contribute in the structural similarity computation. When C is set high, more vertices in G can participate in the process of recursive computation. So P-Rank scores can be accumulated more easily and the convergence therefore will take more time.

We then test how the relative weight, λ, has an impact on P-Rank. As discussed in Section 3.2, λ trades off P-Rank

between the in-link and out-link relationships. When $\lambda = 1$, P-Rank is equal to SimRank. And when $\lambda = 0$, P-Rank is equal to rvs-SimRank. As shown in Figure 11(b), the curve representing $\lambda = 0.5$ lies between curves representing $\lambda = 0$ (rvs-SimRank) and $\lambda = 1$ (SimRank). It means that when $\lambda = 0.5$, P-Rank well balances both in-link and out-link factors for measuring structural similarity. When $\lambda = 0.2$, the out-link relationships are still more important than the in-link ones, and P-Rank is interpolated by similarity scores from both sides. However, the curve representing $\lambda = 0.2$ is quite close to the rvs-SimRank curve ($\lambda = 0$). A similar phenomenon occurs for the curve representing $\lambda = 0.8$, which is quite close to the SimRank curve.

5. CONCLUSION

In this paper we propose a comprehensive structural similarity measure, P-Rank, for large information networks (INs). We start with the basic philosophy of P-Rank that two entities of an IN are similar if (1) they are referenced by similar entities, and (2) they reference similar entities. In comparison with other structural similarity measures, P-Rank takes into account of both in- and out-link relationships of entity pairs and penetrates the structural similarity computation beyond neighborhood of vertices to the entire IN. The advantages of P-Rank are its semantic completeness, robustness and flexibility under different IN settings. P-Rank is shown to be a unified framework for structural similarity measures over massive INs, under which the state-of-the-art similarity measures as CoCitation, Coupling, Amsler and SimRank are all its special cases. We present a fixed point algorithm for computing P-Rank. Efficient pruning techniques under different IN settings are also proposed to reduce the space and time complexity of P-Rank. We perform extensive experimental studies on both real datasets and synthetic datasets and the results confirm the applicability and comprehensiveness of P-Rank, as well as its significant improvement over other structural similarity measures.

6. REFERENCES

[1] R. Amsler. Application of citation-based automatic classification. Technical report, The University of Texas at Austin Linguistics Research Center, December 1972.

[2] I. Antonellis, H. Garcia-Molina, and C.-C. Chang. Simrank++: Query rewriting through link analysis of the click graph. In *Proceedings of VLDB*, pages 408–421, 2008.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[4] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.

[5] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *Fourth SIAM International Conference on Data Mining (SDM' 04)*, April 2004.

[6] D. Fogaras and B. Rácz. Scaling link-based similarity search. In *Proceedings of WWW*, pages 641–650, 2005.

[7] F. Geerts, H. Mannila, and E. Terzi. Relational link-based ranking. In *Proceedings of VLDB*, pages 552–563, 2004.

[8] C. L. Giles. The future of citeseer. In *10th European Conference on PKDD (PKDD'06)*, page 2, 2006.

[9] M. Heymans and A. K. Singh. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics*, 19 Suppl 1, 2003.

[10] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD conference (KDD'02)*, pages 538–543. ACM, 2002.

[11] W. Jiang, J. Vaidya, Z. Balaporia, C. Clifton, and B. Banich. Knowledge discovery from transportation network data. In *Proceedings of the 21st ICDE Conference (ICDE'05)*, pages 1061–1072, 2005.

[12] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis.* John Wiley and Sons, 1990.

[13] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14:10–25, 1963.

[14] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[15] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, and E. Upfal. The web as a graph. In *Proceedings of PODS*, pages 1–10, 2000.

[16] Z. Lin, I. King, and M. R. Lyu. Pagesim: A novel link-based similarity measure for the world wide web. In *Web Intelligence*, pages 687–693, 2006.

[17] D. Lizorkin, P. Velikhov, M. Grinev, and D. Turdakov. Accuracy estimate and optimization techniques for simrank computation. *Proc. VLDB Endow.*, 1(1):422–433, 2008.

[18] A. G. Maguitman, F. Menczer, F. Erdinc, H. Roinestad, and A. Vespignani. Algorithmic computation and approximation of semantic similarity. *World Wide Web*, 9(4):431–456, 2006.

[19] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC'07)*, pages 29–42, 2007.

[20] A. Popescul, G. Flake, S. Lawrence, L. Ungar, and C. L. Giles. Clustering and identifying temporal trends in document databases. In *Proceedings of the IEEE Advances in Digital Libraries*, pages 173–182, 2000.

[21] S. Roy, T. Lane, and M. Werner-Washburne. Integrative construction and analysis of condition-specific biological networks. In *Proceedings of AAAI'07*, pages 1898–1899, 2007.

[22] H. G. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269, 1973.

[23] W. Xi, E. A. Fox, W. Fan, B. Zhang, Z. Chen, J. Yan, and D. Zhuang. Simfusion: Measuring similarity using unified relationship matrix. In *SIGIR*, pages 130–137, 2005.

[24] X. Yin, J. Han, and P. S. Yu. Linkclus: Efficient clustering via heterogeneous semantic links. In *Proceedings of VLDB*, pages 427–438, 2006.

APPENDIX

Proof: [Theorem 3.1]

1. **(*Symmetry*)** According to Equation (3) and Equation (4), it is obvious $R_k(a,b) = R_k(b,a)$ for $k \geq 0$

2. **(*Monotonicity*)** If $a = b$, $R_0(a,b) = R_1(a,b) = \ldots = 1$, so it is obvious that the monotonicity property holds. Let's consider $a \neq b$. According to Equation (3), $R_0(a,b) = 0$. Base on Equation (4), $0 \leq R_1(a,b) \leq 1$. So, $0 \leq R_0(a,b) \leq R_1(a,b) \leq 1$. Let's assume that for all k, $0 \leq R_{k-1}(a,b) \leq R_k(a,b) \leq 1$, then

$$R_{k+1}(a,b) - R_k(a,b) = \lambda \times \frac{C}{|I(a)||I(b)|} \times \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} (R_k(I_i(a), I_j(b)) - R_{k-1}(I_i(a), I_j(b)))$$

$$+ \ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \times \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} (R_k(O_i(a), O_j(b)) - R_{k-1}(O_i(a), O_j(b)))$$

Based on the assumption, we have $(R_k(a,b) - R_{k-1}(a,b)) \geq 0, \forall\, a, b \in G$, so the left hand side $R_{k+1}(a,b) - R_k(a,b) \geq 0$ holds. By induction, we draw the conclusion that for any k, $R_k \leq R_{k+1}$. And based on the assumption, $0 \leq R_k(a,b) \leq 1$, so

$$\lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b))$$
$$\leq \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} 1 = \lambda \times C$$

$$(1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} R_k(O_i(a), O_j(b))$$
$$\leq (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} 1 = (1-\lambda) \times C$$

so, $R_{k+1}(a,b) \leq \lambda C + (1-\lambda)C \leq 1$. By induction, we know that for any k, $0 \leq R_k(a,b) \leq 1$.

3. **(*Existence*)** According to Theorem 3.1-(2), $\forall a, b \in G$, $R_k(a,b)$ is bounded and nondecreasing as k increases. By the Completeness Axiom of calculus, each sequence $R_k(a,b)$ converges to a limit $R(a,b) \in [0,1]$. Note $\lim_{k\to\infty} R_k(a,b) = \lim_{k\to\infty} R_{k+1}(a,b) = R(a,b)$, So we have

$$R(a,b) = \lim_{k\to\infty} R_{k+1}(a,b)$$
$$= \lambda \times \frac{C}{|I(a)||I(b)|} \lim_{k\to\infty} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b))$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \lim_{k\to\infty} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} R_k(O_i(a), O_j(b))$$
$$= \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} \lim_{k\to\infty} R_k(I_i(a), I_j(b))$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} \lim_{k\to\infty} R_k(O_i(a), O_j(b))$$
$$= \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R(I_i(a), I_j(b))$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} R(O_i(a), O_j(b))$$

Note that the limit of $R_k(*,*)$, with respect to k, right satisfies the recursive P-Rank equation, shown in Equation (1).

4. **(*Uniqueness*)** Suppose $s_1(*,*)$ and $s_2(*,*)$ are two solutions to the n^2 iterative P-Rank equations. For any entities $x, y \in G$, let $\delta(x,y) = s_1(x,y) - s_2(x,y)$ be their difference. Let $M = \max_{(x,y)} |\delta(a,b)|$ be the maximum absolute value of any difference. We need to show that $M = 0$. Let $|\delta(x,y)| = M$ for some $a, b \in G$. It is obvious that $M = 0$ if $a = b$. Otherwise,

$$\delta(a,b) = s_1(a,b) - s_2(a,b)$$
$$= \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} (s_1(I_i(a), I_j(b)) - s_2(I_i(a), I_j(b)))$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \times \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} (s_1(O_i(a), O_j(b)) - s_2(O_i(a), O_j(b)))$$

Thus,

$$M = |\delta(a,b)| =$$
$$= \Big|\lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} \delta(I_i(a), I_j(b))$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} \delta(O_i(a), O_j(b))\Big|$$
$$\leq \Big|\lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} \delta(I_i(a), I_j(b))\Big|$$
$$+ \Big|(1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} \delta(O_i(a), O_j(b))\Big|$$
$$\leq \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} |\delta(I_i(a), I_j(b))|$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} |\delta(O_i(a), O_j(b))|$$
$$\leq \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} M$$
$$+ (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} M$$
$$= CM$$

So $M = 0$ when $C \neq 1$. □