An Axiomatic Role Similarity Measure Based on Graph Topology

Weiren Yu[1], Sima Iranmanesh[1], Aparajita Haldar[1], Maoyin Zhang[2], and Hakan Ferhatosmanoglu[1]

[1] University of Warwick, Coventry, CV4 7AL, UK
{weiren.yu, aparajita.haldar, h.ferhatosmanoglu}@warwick.ac.uk
sima89im@gmail.com
[2] Nanjing University of Science and Technology, Jiangsu, China
maoyinzhang@hotmail.com

Abstract. RoleSim and SimRank are popular graph-theoretic similarity measures with many applications in, *e.g.,* web search, collaborative filtering, and sociometry. While RoleSim addresses the automorphic (role) equivalence of pairwise similarity which SimRank lacks, it ignores the neighboring similarity information out of the automorphically equivalent set. Consequently, two pairs of nodes, which are not automorphically equivalent by nature, cannot be well distinguished by RoleSim if the averages of their neighboring similarities over the automorphically equivalent set are the same.
To alleviate this problem: 1) We propose a novel similarity model, namely RoleSim*, which accurately evaluates pairwise role similarities in a more comprehensive manner. RoleSim* not only guarantees the automorphic equivalence that SimRank lacks, but also takes into account the neighboring similarity information outside the automorphically equivalent sets that are overlooked by RoleSim. 2) We prove the existence and uniqueness of the RoleSim* solution, and show its three axiomatic properties (*i.e.,* symmetry, boundedness, and non-increasing monotonicity). 3) We provide a concise bound for iteratively computing RoleSim* formula, and estimate the number of iterations required to attain a desired accuracy. 4) We induce a distance metric based on RoleSim* similarity, and show that the RoleSim* metric fulfills the triangular inequality, which implies the sum-transitivity of its similarity scores. Our experimental results on real and synthetic datasets demonstrate that RoleSim* achieves higher accuracy than its competitors while retaining comparable computational complexity bounds of RoleSim.

1 Introduction

RoleSim is a role-based similarity measure that quantifies the closeness between two objects based on graph topology, with a proliferation of real-life applications [9, 10, 23] in, *e.g.,* link prediction (social network), co-citation analysis (bibliometrics) , motif discovery (bioinformatics), and collaborative filtering (information retrieval). It recursively follows a SimRank-like reasoning that "two nodes

are assessed as role similar if they interact with automorphically equivalent sets of in-neighbors". Intuitively, automorphically equivalent nodes in a graph are objects having similar roles that can be exchanged with minimum effect on the graph structure. Similar to the well-known measure SimRank [7], the recursive nature of RoleSim allows to capture the multi-hop neighboring structures that are automorphically equivalent in a network. Unlike SimRank that measures the similarity of two nodes from the paths connecting them, RoleSim quantifies their similarities through the paths connecting their different roles. As a result, two nodes that are disconnected each other will not be considered as dissimilar by RoleSim if they have similar roles. For evaluating similarity score $s(a, b)$ between nodes a and b, as opposed to SimRank whose similarity $s(a, b)$ takes the average similarity of all the neighboring pairs of (a, b), RoleSim computes $s(a, b)$ by averaging only the similarities over the maximum bipartite matching of all the neighboring pairs of (a, b). This subtle difference enables RoleSim to guarantee the automorphic equivalence, which SimRank lacks, in final scoring results. Therefore, RoleSim has been demonstrated as an effective similarity measure in many real applications. We summarize two of these applications below.

Application 1 (Similarity Search on the Web). Discovering web pages similar to a query page is an important task in information retrieval. In a Web graph, each node represents a web page, and an edge denotes a hyperlink from one page to another. RoleSim can be applied to measure the similarity of two web pages, based on the intuition that "two web pages are role-similar if they are pointed to by the automorphically equivalent sets of their in-neighboring pages". This similarity measure produces more reliable similarity results than the SimRank model [10].

Application 2 (Social Network De-anonymisation). Social network de-anonymisation is a method to validate the strength of anonymisation algorithms that protect a user's privacy. RoleSim has been applied to de-anonymise node mappings based on the similarity information between a crawled network and an anonymised one. Based on the observation that "correct mappings tend to have higher similarity scores", RoleSim iteratively evaluates pairwise node similarities between two networks, and captures the reasoning that "a pair of nodes between two networks is more likely to be a correct mapping if their neighbors are correct mappings". RoleSim has demonstrated superior performance as compared with other existing de-anonymization algorithms [23].

Despite its popularity in real-world applications, RoleSim has a major limitation: with the aim to achieve automorphic equivalence, its similarity score $s(a, b)$ only considers the limited information of the average similarity scores over the automorphically equivalent set (*i.e.,* the maximum bipartite matching) of a's and b's in-neighboring pairs, but neglects the rest of the pairwise in-neighboring similarity information that is out of the automorphically equivalent set. Consequently, RoleSim does not always produce comprehensive similarity results because two pairs of nodes, which are not automorphically equivalent by nature, should be distinguished from each other even though the average values of their in-neighboring similarities over the set of the maximum bipartite matching are the same, as illustrated in Example 1.

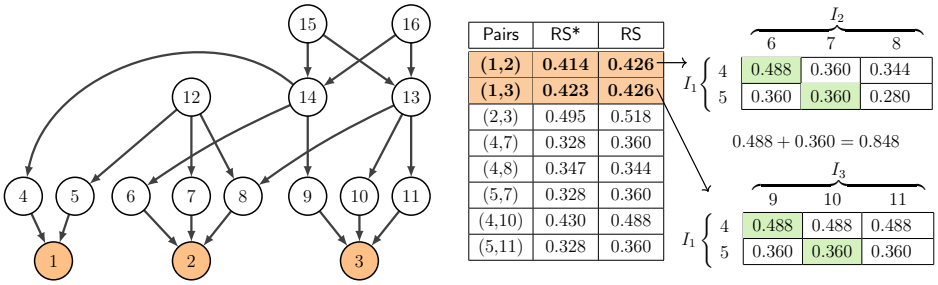


Fig. 1: Limitation of RoleSim (RS) on a tiny web graph, where node-pairs $(1, 2)$ and $(1, 3)$ have the same RoleSim score (0.426) since RS aggregates only the in-neighboring pairs that are automorphically equivalent (colored in green) whose sums are the same ($0.488 + 0.360 = 0.848$), while ignoring the remaining pairs.

Example 1 (Limitation of RoleSim). Consider the web graph G in Figure 1, where each node denotes a web page, and each edge depicts a hyperlink from one page to another. Using RoleSim, we evaluate pairs of similarities between nodes, as partially illustrated in the 'RS' column of the right table. It is discerned that node-pairs $(1, 2)$ and $(1, 3)$ have the same RoleSim similarity values, which is not reasonable. Because node 2 and node 3 are not strictly automorphically equivalent by nature, their similarities with respect to the same query node 1, *i.e.,* $s(1, 2)$ and $s(1, 3)$, should not be the same.

We notice that the main reason why $s(1, 2)$ and $s(1, 3)$ are assessed to be the same by the RoleSim model is that its similarity $s(a, b)$ considers only the average similarity scores over the maximum bipartite matching, denoted as $M_{a,b}$, of (a, b)'s in-neighboring pairs $I_a \times I_b$, where I_a denotes the in-neighbor set of node a, and $\times$ is the Cartesian product of two sets. Thus, the similarity information in the remaining in-neighboring pairs of (a, b), *i.e.,* $I_a \times I_b - M_{a,b}$, are totally ignored. For example, if unfolding the in-neighboring pairs of $(1, 2)$ and $(1, 3)$ respectively, we see that, in the gray cells, $M_{1,2} = \{(4, 6), (5, 7)\}$ (*resp.* $M_{1,3} = \{(4, 9), (5, 10)\}$) is the maximum bipartite matching of $(1, 2)$'s (*resp.* $(1, 3)$'s) in-neighboring pairs $I_1 \times I_2$ (*resp.* $I_1 \times I_3$). The sum of the similarity values over $M_{1,2}$ is $0.488 + 0.360 = 0.848$, which is the same as that over $M_{1,3}$. Thus, RoleSim cannot distinguish $s(1, 2)$ from $s(1, 3)$. □

Example 1 illustrates that, to effectively evaluate $s(a, b)$, relying only on the in-neighboring-pairs similarities in the maximum bipartite matching $M_{a,b}$ (*e.g.,* RoleSim) is not enough. Although RoleSim has the advantage of finding the most influential pairs $M_{a,b}$ among all the in-neighboring pairs $I_a \times I_b$ for achieving automorphic equivalence, it *completely* ignores the similarity information outside $M_{a,b}$. For instance in Example 1, there are opportunities to take good advantage of the similarity values in the regions $I_1 \times I_2 - M_{1,2}$ and $I_1 \times I_3 - M_{1,3}$ which would be helpful to distinguish $s(1, 2)$ from $s(1, 3)$ further when the average similarities over $M_{1,2}$ and $M_{1,3}$ are the same.

Contributions. Motivated by this, our main contributions are as follows:

1) We first propose a novel similarity model, RoleSim*, which accurately evaluates pairwise role similarities in a more comprehensive fashion. Compared

with the existing well-known similarity models (*e.g.,* SimRank and RoleSim), RoleSim* not only guarantees the automorphic equivalence that SimRank lacks, but also takes into consideration the pairwise similarities outside the automorphically equivalent sets that are overlooked by RoleSim. (Section 3.1)

2) We prove the existence and uniqueness of the RoleSim* solution, and show three key axiomatic properties of RoleSim*, *i.e.,* symmetry, boundedness, and non-increasing monotonicity of its iterative similarity scores. (Section 3.2)

3) We derive an iterative formula for computing RoleSim* similarities, and a concise upper bound is obtained, which can estimate the total number of iterations required for attaining a desired accuracy. (Sections 3.3 and 3.4)

4) We induce a distance metric based on our RoleSim* measure, and rigorously show that the RoleSim* distance metric fulfills the triangular inequality which other measures (*e.g.,* cosine distance) lack. This implies the sum-transitivity of the RoleSim* measure. (Section 3.5)

5) We conduct an experimental study to validate the effectiveness of our RoleSim* model. Our empirical results show that RoleSim* achieves higher accuracy than the existing competitors (*e.g.,* RoleSim and SimRank) while entailing comparable computational complexity bounds of RoleSim. (Section 4)

2 Related Work

Graph-based similarity models have been popular since SimRank measure was proposed by Jeh and Windom [7]. SimRank is a node-pair similarity measure, which follows the recursive idea that "two nodes are considered as similar if they are pointed to by similar nodes". Since then, there have been surges of studies focusing on optimization problems to accelerate SimRank computation as the naive SimRank computing method entails quadratic time in the number of nodes. According to assumptions on data updates, recent results can be divided into static algorithms [4, 11, 20, 24, 28, 36, 27, 1, 5, 15, 31], and dynamic algorithms on evolving graphs [34, 25, 12, 22, 8, 18, 30]. According to types of queries, these results are classified into single-source SimRank [8, 11, 24, 34, 18], single-pair SimRank [14, 6], all-pairs SimRank [29, 28, 1, 19], and partial-pairs SimRank [20, 33].

There are many studies on semantic problems of pairwise similarity measures. Various SimRank-like measures have come into play, including C-Rank [26], SimFusion [32], P-Rank [35], RoleSim [9], MatchSim [17], ASCOS [2], SimRank* [31], CoSimRank [21], SemSim [27]. Among them, RoleSim has stood out as a promising role-based similarity model, due to its elegant intuition that "if two nodes are automorphically equivalent, they should share the same role and their role similarity should be maximal". To speed up the RoleSim computation, an approximate heuristic, named Iceberg RoleSim, was devised to prune small similarity values below a threshold.

Unlike SimRank that takes the average similarity of all the neighboring pairs of (a, b), RoleSim computes $s(a, b)$ by averaging only the similarities over the maximum bipartite matching $M_{a,b}$. However, all the similarity information not included in the matching $M_{a,b}$ is completely ignored by RoleSim. In contrast, our RoleSim* model can effectively capture these information while guaranteeing automorphic equivalence.

There have also been a host of studies on variations of RoleSim [17, 13, 23, 3]. Lin *et al.* [17] introduced MatchSim whose similarity is defined to be the average similarity of (a, b)'s maximum matched neighbors. It differs from RoleSim in that MatchSim initialises $s_0(a, b) = 1$ if $a = b$, and 0 otherwise, whereas RoleSim initialises all $s_0(*, *) = 1$. As a result, MatchSim scores do not guarantee automorphic equivalence. Li *et al.* [13] proposed CentSim, a centrality based role similarity measure, which compares the centrality values of two nodes to evaluate their similarity. This measure employs several types of centralities including PageRank, Degree and Closeness for each node, and considers the weighted average of them for evaluating CentSim scores. Recently, Shao *et al.* [23] introduced RoleSim++, an extension of RoleSim, which considers both incoming and outgoing neighbors in a digraph for social network de-anonymisation. It employs a novel matching algorithm, called NeighborMatch, to find matching for inner and outer neighbors, respectively. Furthermore, a threshold based version, α-RoleSim++, is proposed to eliminate tiny scores for speedup further. Most recently, Chen *et al.* [3] suggest a scalable model, StructSim, with an efficient BinCount matching algorithm and present a hierarchical scheme, which achieves a more efficient role similarity computation.

3 RoleSim*

3.1 RoleSim* Formulation

The central intuition underpinning RoleSim* follows a recursive concept that "two distinct nodes are assessed to be similar if they

1. *mainly interact with the automorphically equivalent sets of in-neighbors, and*
2. *are in-linked by similar nodes that are out of automorphically equivalent sets.*

The starting point for this recursion is to assign each pair of nodes a similarity score 1, meaning that initially no pairs of nodes are thought of to be more (or less) similar than others.

Notations. Before illustrating the mathematical definition to reify the RoleSim* intuition, we introduce the following notations.

Let $G = (V, E)$ be a directed graph with a set of nodes V and a set of edges E. Let I_a be all in-neighbors of node a, and $|I_a|$ the cardinality of the set I_a. For a pair of nodes (a, b) in G, we denote by $I_a \times I_b = \{(x, y) \mid \forall x \in I_a \text{ and } \forall y \in I_b\}$ all in-neighboring pairs of (a, b), and $s(a, b)$ the RoleSim* similarity score between nodes a and b. Using $I_a \times I_b$ and $s(a, b)$, we define a weighted complete bipartite graph, denoted by $K_{|I_a|,|I_b|} = (I_a \cup I_b, I_a \times I_b)$, with each edge $(x, y) \in I_a \times I_b$ carrying the weight $s(a, b)$. We denoted by $M_{a,b}$ $(\subseteq I_a \times I_b)$ the maximum weighted matching in bipartite graph $K_{|I_a|,|I_b|}$.

Example 2. Recall digraph G in Figure 1. For nodes 1 and 2, their in-neighbors sets are $I_1 = \{4, 5\}$ and $I_2 = \{6, 7, 8\}$, respectively. The set of all in-neighboring pairs of $(1, 2)$ is $I_1 \times I_2 = \{(\mathbf{4}, \mathbf{6}), (4, 7), (4, 8), (5, 6), (\mathbf{5}, \mathbf{7}), (5, 8)\}$. The maximum matching of bipartite graph $(I_1 \cup I_2, I_1 \times I_2)$ is $M_{1,2} = \{(4, 6), (5, 7)\}$ (bold). □

Symbol	Description
G	directed graph $G = (V, E)$ with a set nodes V and a set of edges E
I_a	all in-neighbors of node a in G
$\|I_a\|$	cardinality of the set I_a (*i.e.,* the number of nodes in I_a)
$M_{a,b}$	maximum weighted matching in bipartite graph $K_{\|I_a\|,\|I_b\|} = (I_a \cup I_b, I_a \times I_b)$
$s(a,b)$	RoleSim* similarity score between nodes a and b
β	damping factor $(0 < \beta < 1)$
λ	relative weight that balances similarities inside and outside $M_{a,b}$ $(0 < \lambda < 1)$
K	total number of iterations

Table 1: Description of Main Symbols

Other notations frequently used throughout this paper are listed in Table 1.

RoleSim* Formula. Based on our aforementioned intuition, we formally formulate the RoleSim* model as follows:

$$s(a,b) = \beta \times \Bigg(\lambda \times \overbrace{\frac{1}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y) \in M_{a,b}} s(x,y)}^{\text{Part 1: average similarity over maximum matching } M_{a,b}} + (1-\lambda) \times \underbrace{\frac{1}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y) \in (I_a \times I_b) - M_{a,b}} s(x,y)}_{\text{Part 2: average similarity over } (I_a \times I_b) - M_{a,b}} \Bigg) + (1-\beta) \tag{1}$$

In Eq.(1), for every pair of nodes (a, b), the set of their in-neighboring pairs, $I_a \times I_b$, is split into two subsets: $I_a \times I_b = M_{a,b} \cup (I_a \times I_b - M_{a,b})$. As a result, the definition of RoleSim* consists of two parts: Part 1 is the average similarity over maximum matching $M_{a,b}$, indicating the contribution from (a, b) interacting with the automorphically equivalent set, $M_{a,b}$, of (a, b)'s in-neighbors pairs. Part 2 is the average similarity over $(I_a \times I_b) - M_{a,b}$, corresponding to the contribution from (a, b) being pointed to by the rest of (a, b)'s in-neighbors pairs out of automorphically equivalent set $M_{a,b}$. The relative weight of Part 1 and 2 is balanced by a user-controlled parameter $\lambda \in [0, 1]$. β is a damping factor between 0 and 1, which is often set to 0.6 or 0.8, implying that similarity propagation made with distant in-neighbors is penalised by an attenuation factor β across edges. When I_a (or I_b) $= \emptyset$, which implies the maximum matching $M_{a,b} = \emptyset$, we define Part 1 and Part 2$= 0$ in order to avoid the denominators of the fraction in Part 1 and 2 being zeros.

Fixed-Point Iteration. To solve RoleSim* similarity $s(a, b)$ in Eq.(1), we adopt the following fixed-point iterative scheme:

$$s_0(a,b) = 1 \qquad (\forall a, b) \tag{2}$$

$$s_{k+1}(a,b) = \beta \times \Bigg(\frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y) \in M_{a,b}} s_k(x,y) + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y) \in (I_a \times I_b) - M_{a,b}} s_k(x,y) \Bigg) + (1-\beta) \tag{3}$$

where $s_k(a,b)$ denotes the RoleSim* score between nodes a and b at iteration k. Based on Eqs.(2) and (3), we can iteratively compute all pairs of similarity scores $s_{k+1}(*,*)$ from those at the last iteration $s_k(*,*)$. The fixed-point scheme in Eqs.(2) and (3) implies an iterative algorithm for RoleSim* computation, which requires $O(K|E|^2)$ time to compute $|V|^2$ node-pairs for K iterations.

Threshold-Based RoleSim*. To accelerate RoleSim* computation in Algorithm 1, we notice that there are a significant number of node pairs whose iterative similarity values $s_k(*,*)$ are very close to their convergent scores $s(*,*)$ and thus will not change much in subsequent iterations. We eliminate such pairs from unnecessary iterative computations, based on Cauchy Convergence Criterion:

$$\lim_{k\to+\infty} s_k(*,*) = s(*,*) \;\Leftrightarrow\; \exists\delta \;\; s.t. \quad |s_k(*,*) - s_{k+1}(*,*)| < \delta$$

Hence, we propose the following threshold-based RoleSim* model, where δ is a user-controlled threshold, which is a speed-accuracy trade-off.

$$s_0^\delta(a,b) = 1$$

$$s_{k+1}^\delta(a,b) = \begin{cases} s_k^\delta(a,b) & \text{if } s_{k-1}^\delta(a,b) - s_k^\delta(a,b) < \delta \\ 1-\beta & \text{if } s_k^\delta(a,b) < (1-\beta)+\delta \\ \beta \times \left(\frac{\lambda}{|I_a|+|I_b|-|M_{a,b}|} \sum\limits_{(x,y)\in M_{a,b}} s_k^\delta(x,y) \right. & \text{otherwise} \\ \left. + \frac{1-\lambda}{|I_a|\times|I_b|-|M_{a,b}|} \sum\limits_{(x,y)\in (I_a\times I_b)-M_{a,b}} s_k^\delta(x,y) \right) + (1-\beta) & \end{cases}$$

3.2 Axiomatic Properties for RoleSim* Iterative Similarity

Based on the definition of iterative similarity $s_k(a,b)$ in Eqs.(2) and (3), we next show three axiomatic properties of RoleSim*, *i.e.,* symmetry, boundedness, and non-increasing monotonicity, based on the following theorem.

Theorem 1. *The iterative RoleSim* $\{s_k(a,b)\}$ in Eqs.*(2) *and* (3) *have the following key properties: for any node pair* (a,b) *and each iteration* $k = 0,1,\cdots,$

1. ***(Symmetry)*** $s_k(a,b) = s_k(b,a)$
2. ***(Boundedness)*** $1-\beta \le s_k(a,b) \le 1$
3. ***(Monotonicity)*** $s_{k+1}(a,b) \le s_k(a,b)$

Theorem 1 indicates that, for every iteration $k = 0,1,2,\cdots$, $\{s_k(a,b)\}$ is a bounded symmetric scoring function. Moreover, as $k \to \infty$, it can be readily verified that the exact solution $s(a,b)$ also is a bounded symmetric measure, which is similar to SimRank and RoleSim. In comparison, other measures (*e.g.,* Hitting Time and Random Walk with Restart) are asymmetric ones.

3.3 Existence & Uniqueness

It is worth mentioning that, as opposed to SimRank whose iterative similarity is non-decreasing between 0 and 1 *w.r.t.* k, RoleSim* similarity is non-increasing between $1-\beta$ and 1. The bounded non-increasing property of RoleSim* guarantees the existence and uniqueness of its exact solution $s(a,b)$, as shown below:

Theorem 2 (Existence and Uniqueness). *There always exists a unique solution $s(a,b)$ (i.e., the exact RoleSim score) to Eqs.*(2) *and* (3) *such that the iterative RoleSim similarity $\{s_k(a,b)\}$ converges to it, i.e., $\lim_{k\to\infty} s_k(a,b) = s(a,b)$.*

3.4 Accuracy Estimation

Having proved the existence and uniqueness of the exact RoleSim* solution, we are now ready to investigate the error bound of the difference between the k-th iterative similarity $s_k(a,b)$ and exact one $s(a,b)$. In virtue of the non-increasing monotonicity of $\{s_k(a,b)\}$, one can readily show that the exact $s(a,b)$ is the lower bound of all the iterative similarities $\{s_k(a,b)\}$, *i.e.,* $s_k(a,b) \geq s(a,b)$ $(\forall k)$. The following theorem further provides a concise upper bound to measure the closeness between $s_k(a,b)$ and $s(a,b)$.

Theorem 3 (Iterative Error Bound). *For every iteration number $k = 0, 1, 2, \cdots$, the difference between $s_k(a,b)$ and $s(a,b)$ is bounded by*

$$s_k(a,b) - s(a,b) \leq \beta^{k+1} \qquad (\forall a, b) \tag{4}$$

Theorem 3 derives a concise exponential upper bound for the difference between the k-th iterative similarity $s_k(a,b)$ and exact $s(a,b)$. Combining this bound with the non-increasing monotonicity $s_k(a,b) \geq s(a,b)$, we can obtain that the k-th iterative error $s_k(a,b) - s(a,b)$ is between 0 and β^{k+1}. Moreover, Theorem 3 also implies that, given desired accuracy $\epsilon > 0$, the total number of iterations required for computing RoleSim* similarity is $K = \lceil \log_\beta \epsilon \rceil$.

3.5 "Sum-Transitivity" of RoleSim* Similarity

In this section, we investigate the transitive property of the proposed RoleSim* similarity measure. Intuitively, when a similarity measure $s(*,*)$ fulfils the transitive property, it means that, for any three nodes a, b, c in the graph, if a is similar to b and b is similar to c, it implies that a is likely to be similar to c. The transitivity feature is useful in many real applications, *e.g.,* for predicting and recommending links in a graph.

To study the transitive property of RoleSim*, let us induce a distance $d(a,b) := 1 - s(a,b)$ from the RoleSim* measure. Due to $s(*,*) \in [1-\beta, 1]$, the distance $d(*,*)$ is between 0 and β. In what follows, we will show that $d(*,*)$ satisfies the triangular inequality, which is an indication of $s(*,*)$ transitivity.

We first show the following lemma, which is needed for further proof of RoleSim* triangular inequality.

Lemma 1. *Let $s_k(*,*)$ be the k-th iterative RoleSim* similarity to Eqs.*(2) *and* (3)*. For any three nodes a, b, c in a graph, if $s_k(a,b) + s_k(b,c) - s_k(a,c) \leq 1$ holds at iteration k, the following inequalities holds:*

$$P_1 := \frac{\sum\limits_{(x,y)\in M_{a,b}} s_k(x,y)}{|I_a| + |I_b| - |M_{a,b}|} + \frac{\sum\limits_{(y,z)\in M_{b,c}} s_k(y,z)}{|I_b| + |I_c| - |M_{b,c}|} - \frac{\sum\limits_{(x,z)\in M_{a,c}} s_k(x,z)}{|I_a| + |I_c| - |M_{a,c}|} \leq 1 \tag{5}$$

$$P_2 := \frac{\sum_{(x,y)\in(I_a\times I_b)-M_{a,b}} s_k(x,y)}{|I_a|\times|I_b|-|M_{a,b}|} + \frac{\sum_{(y,z)\in(I_b\times I_c)-M_{b,c}} s_k(y,z)}{|I_b|\times|I_c|-|M_{b,c}|} - \frac{\sum_{(x,z)\in(I_a\times I_c)-M_{a,c}} s_k(x,z)}{|I_a|\times|I_c|-|M_{a,c}|} \leq 1 \quad (6)$$

Leveraging Lemma 1, we are now ready to show the sum-transitivity of the RoleSim* similarity distance, which is the main result in this subsection:

Theorem 4. *The RoleSim* similarity $s(a,b)$ defined by Eq.(1) satisfies the following "sum-transitive" property: Let $d(a,b) := 1 - s(a,b)$ be the closeness between nodes a and b. Then, for any three nodes a, b, c in a graph, the following triangular inequality holds, i.e.,*

$$d(a,b) + d(b,c) \geq d(a,c) \quad (7)$$

4 Experimental Evaluation

4.1 Experimental Settings

Datasets. We use both real and synthetic datasets, as illustrated below:

Datasets	Abbr.	#Node-Pairs	#Nodes	#Edges	Type
Amazon	(AMZ)	25,867,396	5,086	8,970	Directed
DBLP	(DBLP)	5,626,384	2,372	7,106	Undirected
Synthetic	(SYN)	4,000,000	2,000	5,481	Undirected

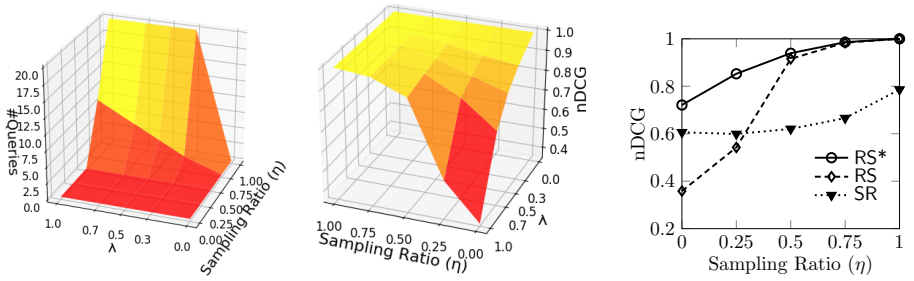
- Amazon. A co-purchasing digraph crawled from *Customers Who Bought This Item Also Bought* feature of Amazon[3]. Each node is a product, and edge $i \to j$ means that product j appears in the frequent co-purchasing list of i.
- DBLP. A collaboration (undirected) graph taken from DBLP bibliography.[4] We extract a co-authorship subgraph from six top conferences in computer science (SIGMOD, VLDB, PODS, KDD, SIGIR, ICDE) during 2018–2020. If two authors (nodes) co-authored a paper, there is an edge between them.
- Synthetic. A random scale-free graph with a power-law degree distribution, generated by GenRndPowerLaw function in C++ SNAP Library.[5]

All experiments are conducted on a PC with Intel Core i7-10510U 2.30GHz CPU and 16GB RAM, using Windows 8 Professional 64-bit. Each experiment is repeated 5 times and the average is reported.

Compared Algorithms. We implemented all the following algorithms in VC++:

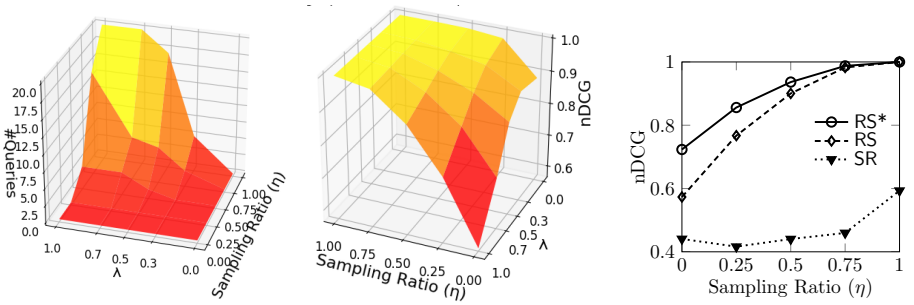
Models	Abbr.	Description
RoleSim*	(RS*)	our proposed RoleSim* model in Algorithm 1.
SimRank	(SR)	a pairwise similarity model proposed by Jeh and Widom [7].
MatchSim	(MS)	a similarity model relying on the matched neighbors of node pairs [16].
RoleSim	(RS)	a model that guarantees the automorphically equivalence of nodes [9].
RoleSim++	(RS++)	an enhanced RoleSim that considers both in- and out-neighbors [23].
CentSim	(CS)	a similarity model that compares the centrality values of node pairs [13].

[3] www.amazon.co.uk
[4] www.informatik.uni-trier.de/~ley/db/
[5] https://snap.stanford.edu/data/index.html



(a) #Queries in Top-5 RS* (b) nDCG RS* (c) nDCG $\lambda = 0.7$

Fig. 2: Effect of Sampling Ratio (η) and Weight (λ) on Ranking Quality (DBLP)



(a) #Queries in Top-5 RS* (b) nDCG RS* (c) nDCG $\lambda = 0.7$

Fig. 3: Effect of Sampling Ratio (η) and Weight (λ) on Ranking Quality (AMZ)

Parameters. We use the following parameters as default: (a) damping factor $\beta = 0.8$, (b) relative weight $\lambda = 0.7$, (c) total number of iterations $K = 4$.

Semantic Evaluation. We design an unsupervised evaluation setting to quantify the effectiveness of the similarity measures in preserving self-similarity under different conditions. In particular, we study the effect of sampling the immediate neighborhood of a query point on similarity scores in RoleSim* compared with SimRank and RoleSim. Consider a single query node q. In our experiment, we create a node q' and add it to the graph. We connect q' to some proportion (η) of the total number of neighbors of q, and hereby refer to q' as the "sampled clone". The similarity scores of q to all other points in the graph are computed using SimRank, RoleSim, and RoleSim*. We evaluate how much the relative similarities are preserved when different measures are used. We vary η in q' with step size 0.25 (and ensuring no orphaned nodes), and additionally consider $\lambda = 0.0, 0.3, 0.5, 0.7, 1.0$ for RoleSim*. Our results are aggregated over 20 queries on DBLP and AMZ graphs respectively, where query nodes are chosen as having high degree of neighbors.

#	RS*($\lambda = 0.6$)	RS*($\lambda = 0.8$)	RS	SR
1	Nitesh V. Chawla	Xia Hu	Xia Hu	Yuan Fang
2	Danai Koutra	Nitesh V. Chawla	Nitesh V. Chawla	Chenwei Zhang
3	Yanjie Fu	Yanjie Fu	Yanjie Fu	Nan Du
4	**Jure Leskovec**	**Jure Leskovec**	**Huan Liu**	Wei Fan
5	Haifeng Chen	Danai Koutra	**Jure Leskovec**	Lichao Sun
6	Xia Hu	Haifeng Chen	Haifeng Chen	Weiran Huang
7	Xing Xie	Xing Xie	Danai Koutra	Jianxin Ma
8	Xiangnan He	Xiangnan He	Xing Xie	Xinyue Liu
9	Di Niu	Di Niu	Xiangnan He	Binbin Hu
10	Jennifer G. Dy	**Huan Liu**	Fenglong Ma	Daixin Wang
...	...	...	...	...
28	**Huan Liu**	Dawei Yin	Di Wu	Ning Wu
...	...	...	...	...
89	Xiang Li	Han Zhu	Qinyong Wang	**Huan Liu**
...	...	...	...	...
350	Mao Yang	Houdong Hu	Xi (Stephen) Chen	**Jure Leskovec**

Table 2: Similarity rankings for "Philip S. Yu" on DBLP co-authorships data

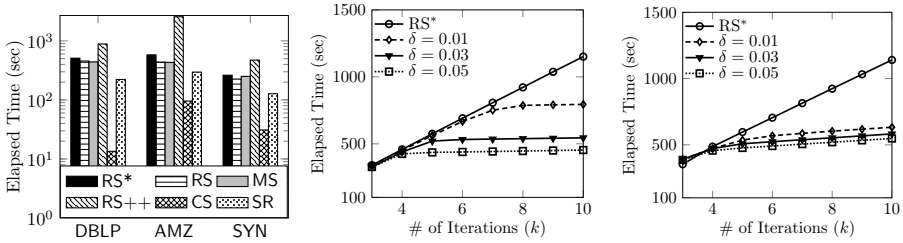
4.2 Experimental Results

Semantic Accuracy. We first count the number of queries where the sampled clone q' appears in the top-k ($k = 1, 5, 10$) similar nodes to query q for RoleSim*. Intuitively, this studies how much structural information is gleaned about a query node. Figure 2(a) presents the number of such queries out of 20 on the undirected DBLP graph, considering top-5 similarity scores. Other top-k plots are omitted, but show that with increasing k for a given sampling proportion there are more such queries even at lower λ.

Next, we test the impact of sampling η and λ on ranking quality in RoleSim*. We plot the average ranking quality (normalized discounted cumulative gain (nDCG)), considering top-100 similar nodes of the sampled clone and comparing this to the baseline original query. We observe that the trend (with respect to η) seen in Figure 2(b) and Figure 3(b) for $\lambda = 1$ resembles that for RoleSim, and the trend for $\lambda = 0.5$ is close to that for SimRank.

Finally, we consider a fixed value of $\lambda = 0.7$ and confirm that the RoleSim* has higher ranking quality compared to SimRank and RoleSim, with respect to the average nDCG. Figure 2(c) with undirected DBLP graph shows that RoleSim* produces a more consistent nDCG even with small η. For the directed AMZ graph in Figure 3(c) too, RoleSim shows significant improvement at lower sampling, and the performance of SimRank is negatively affected throughout, while RoleSim* remains stable.

Qualitative Case Study. Table 2 compares the similarity ranking results from three algorithms (SR, RS and RS*) for retrieving top-10 most similar authors *w.r.t.* query "Philip S. Yu" on DBLP. From the results, we see that the top rankings of RS* are similar to RS, highlighting its capability to effectively capture automorphic equivalent neighboring information. For instance, "Jure Leskovec" is top-ranked in RS* list. This is reasonable because he and "Philip S. Yu" have similar roles - they are both Professors in Computer Science with close research expertise (*e.g.,* knowledge discovery, recommender systems, commonsense reasoning). However, the rankings of RS* are different from those of RS. For example, "Jure Leskovec" is ranked 350^{th} by SR, but 4^{th} by RS* and RS.



(a) Varying Datasets (b) Varying k on DBLP (c) Varying k on AMZ

Fig. 4: Elapsed Time Comparison for Different Threshold-Based RS*

This is because SimRank can only capture connected paths between two authors while ignoring their automorphic equivalent structure. "Jure Leskovec" has rare collaborations with "Philip S. Yu", both direct and indirect, thus leading to a low SimRank score.

To evaluate RS* further, we choose two different values for $\lambda \in \{0.6, 0.8\}$ to show how RS* ranking results are perturbed *w.r.t.* λ. From the results, we notice that, when λ is varied from 0.6 to 0.8, nodes with small SR scores (*e.g.,* "Jure Leskovec") exhibit a stable position in RS* ranking, whereas nodes having higher SR scores (*e.g.,* "Huan Liu") have a substantial change. This conforms with our intuition because "Huan Liu"'s collaboration with "Philip S. Yu" is closer than "Jure Leskovec"'s, and RS* is able to capture both connectivity and automorphic equivalence of two authors using a balanced weight λ. Thus, compared with "Jure Leskovec", "Huan Liu" who has higher SimRank value with "Philip S. Yu" is more sensitive to λ change, as expected.

Computational Time. Figure 4(a) compares the computational time of six algorithms (RS*, RS, MS, RS++, CS, SR) on various datasets (AMZ, DBLP, SYN), respectively. We notice that, on each dataset, RS* has comparable computational time to RS and MS. This implies that RS* achieves high accuracy without sacrificing running speed. In addition, RS*, RS, and MS are 2–4 times faster than RS++. This is because RS* need to find two maximum bipartite matchings for both in- and out-neighboring pairs, as opposed to RS* that involves the computation of only one matching. SR is slightly slower than RS*. This is consistent with our analysis as SR simply takes the average of all similarities of the in-neighboring pairs without the need to find the maximum bipartite matching. CS achieves the fastest speed since it simply assesses a node-pair similarity by aggregating their centrality values, thereby leading to low accuracy.

Figures 4(b) and 4(c) show the effect of iteration number k and threshold δ on the running time of RS* on DBLP and AMZ, respectively. For each dataset, we vary δ from 0 to 0.05. When $\delta = 0$, it reduces to RS* algorithm. From the results on both datasets, we discern that, for each fixed δ, the running time of threshold-based RS* increases as k grows. When δ becomes larger, the growth rate of RS* time tends to be sublinear. For example, when $\delta = 0.05$ on DBLP, only after $k = 5$ iterations, the increasing time of threshold-based RS* has leveled off. In contrast, when $\delta = 0.01$, the time becomes steady after $k = 8$ iterations. The reason is that a higher setting of threshold δ implies a larger number of

pairs to be pruned per iteration, thus leading to the growth rate of the running time decreasing in an earlier stage during iterations.

5 Conclusion

We propose RoleSim*, a novel similarity model that guarantees automorphic equivalence and also considers neighboring similarity information beyond automorphically equivalent sets, thereby achieving better performance than both SimRank and RoleSim. We prove the existence and uniqueness of the RoleSim* solution, show that iteratively computing RoleSim* is bounded, and induce a RoleSim* distance obeying sum-transitivity of similarity scores. We also evaluate our model on DBLP, AMZ, and SYN datasets to demonstrate its superior ranking quality and comparable complexity to competitors.

References

1. I. Antonellis, H. Garcia-Molina, and C.-C. Chang. SimRank++: Query rewriting through link analysis of the click graph. *PVLDB*, 1(1), 2008.
2. H. Chen and C. L. Giles. ASCOS++: an asymmetric similarity measure for weighted networks to address the problem of simrank. *ACM Trans. Knowl. Discov. Data*, 10(2):15:1–15:26, 2015.
3. X. Chen, L. Lai, L. Qin, and X. Lin. StructSim: Querying structural node similarity at billion scale. In *ICDE*, pages 1950–1953, 2020.
4. Y. Fujiwara, M. Nakatsuji, H. Shiokawa, and M. Onizuka. Efficient search algorithm for SimRank. In *ICDE*, pages 589–600, 2013.
5. G. He, H. Feng, C. Li, and H. Chen. Parallel SimRank computation on large graphs with iterative aggregation. In *KDD*, 2010.
6. J. He, H. Liu, J. X. Yu, P. Li, W. He, and X. Du. Assessing single-pair similarity over graphs by aggregating first-meeting probabilities. *Inf. Syst.*, 42:107–122, 2014.
7. G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In *KDD*, pages 538–543, 2002.
8. M. Jiang, A. W. Fu, R. C. Wong, and K. Wang. READS: A random walk approach for efficient and accurate dynamic simrank. *PVLDB*, 10(9):937–948, 2017.
9. R. Jin, V. E. Lee, and H. Hong. Axiomatic ranking of network role similarity. In C. Apté, J. Ghosh, and P. Smyth, editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 922–930. ACM, 2011.
10. R. Jin, V. E. Lee, and L. Li. Scalable and axiomatic ranking of network role similarity. *TKDD*, 8(1):3:1–3:37, 2014.
11. M. Kusumoto, T. Maehara, and K. Kawarabayashi. Scalable similarity search for SimRank. In *SIGMOD*, pages 325–336, 2014.
12. C. Li, J. Han, G. He, X. Jin, Y. Sun, Y. Yu, and T. Wu. Fast computation of SimRank for static and dynamic information networks. In *EDBT*, 2010.
13. L. Li, L. Qian, V. E. Lee, M. Leng, M. Chen, and X. Chen. Fast and accurate computation of role similarity via vertex centrality. In J. Li and Y. Sun, editors, *WAI*, volume 9098 of *Lecture Notes in Computer Science*, pages 123–134. Springer, 2015.
14. P. Li, H. Liu, J. X. Yu, J. He, and X. Du. Fast single-pair simrank computation. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 571–582. SIAM, 2010.

15. Z. Li, Y. Fang, Q. Liu, J. Cheng, R. Cheng, and J. C. S. Lui. Walking in the cloud: Parallel SimRank at scale. *PVLDB*, 9(1):24–35, 2015.
16. Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Detecting splogs via temporal dynamics using self-similarity analysis. *TWEB*, 2(1):4:1–4:35, 2008.
17. Z. Lin, M. R. Lyu, and I. King. MatchSim: a novel similarity measure based on maximum neighborhood matching. *Knowl. Inf. Syst.*, 32(1):141–166, 2012.
18. Y. Liu, B. Zheng, X. He, Z. Wei, X. Xiao, K. Zheng, and J. Lu. ProbeSim: Scalable single-source and top-k simrank computations on dynamic graphs. *PVLDB*, 11(1):14–26, 2017.
19. D. Lizorkin, P. Velikhov, M. N. Grinev, and D. Turdakov. Accuracy estimate and optimization techniques for SimRank computation. *VLDB J.*, 19(1), 2010.
20. T. Maehara, M. Kusumoto, and K. Kawarabayashi. Scalable simrank join algorithm. In *ICDE*, pages 603–614, 2015.
21. S. Rothe and H. Schütze. CoSimRank: A flexible & efficient graph-theoretic similarity measure. In *ACL*, pages 1392–1402. The Association for Computer Linguistics, 2014.
22. Y. Shao, B. Cui, L. Chen, M. Liu, and X. Xie. An efficient similarity search framework for SimRank over large dynamic graphs. *PVLDB*, 8(8):838–849, 2015.
23. Y. Shao, J. Liu, S. Shi, Y. Zhang, and B. Cui. Fast de-anonymization of social networks with structural information. *Data Science and Engineering*, 4(1):76–92, 2019.
24. B. Tian and X. Xiao. SLING: A near-optimal index structure for SimRank. In *SIGMOD*, pages 1859–1874, 2016.
25. Y. Wang, X. Lian, and L. Chen. Efficient simrank tracking in dynamic graphs. In *ICDE*, pages 545–556, 2018.
26. S. Yoon, S. Kim, and S. Park. C-Rank: A link-based similarity measure for scientific literature databases. *Inf. Sci.*, 326:25–40, 2016.
27. B. Youngmann, T. Milo, and A. Somech. Boosting SimRank with semantics. In *EDBT*, pages 37–48, 2019.
28. W. Yu, X. Lin, and W. Zhang. Towards efficient SimRank computation on large networks. In *ICDE*, pages 601–612, 2013.
29. W. Yu, X. Lin, W. Zhang, and J. A. McCann. Fast all-pairs SimRank assessment on large graphs and bipartite domains. *IEEE Trans. Knowl. Data Eng.*, 27(7):1810–1823, 2015.
30. W. Yu, X. Lin, W. Zhang, and J. A. McCann. Dynamical SimRank search on time-varying networks. *VLDB J.*, 27(1):79–104, 2018.
31. W. Yu, X. Lin, W. Zhang, J. Pei, and J. A. McCann. SimRank*: Effective and scalable pairwise similarity search based on graph topology. *VLDB J.*, 28(3):401–426, 2019.
32. W. Yu, X. Lin, W. Zhang, Y. Zhang, and J. Le. SimFusion+: Extending SimFusion towards efficient estimation on large and dynamic networks. In *SIGIR*, pages 365–374, 2012.
33. W. Yu and J. A. McCann. Efficient partial-pairs SimRank search for large networks. *PVLDB*, 8(5):569–580, 2015.
34. W. Yu and F. Wang. Fast exact CoSimRank search on evolving and static graphs. In *WWW*, pages 599–608, 2018.
35. P. Zhao, J. Han, and Y. Sun. P-Rank: A comprehensive structural similarity measure over information networks. In *CIKM*, 2009.
36. R. Zhu, Z. Zou, and J. Li. Simrank computation on uncertain graphs. In *ICDE*, pages 565–576, 2016.

A Proofs of Theorems & Lemmas

A.1 Proof of Theorem 1

Proof. 1. **(Symmetry)** By virtue of Eqs.(2) and (3), $s_k(a,b) = s_k(b,a)$ follows immediately.

2. **(Boundedness)** We will prove by induction on k. For $k = 0$, it is apparent that $s_0(a,b) = 1 \in [1-\beta, 1]$. For $k > 0$, we assume that $s_k(x,y) \leq 1$ holds, and will prove that $s_{k+1}(x,y) \leq 1$ holds as follows: Let

$$P_1 := \frac{1}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y)\in M_{a,b}} \underbrace{s_k(x,y)}_{\leq 1} \leq \frac{|M_{a,b}|}{|I_a| + |I_b| - |M_{a,b}|} = \frac{\min\{|I_a|, |I_b|\}}{\max\{|I_a|, |I_b|\}} \leq 1$$

$$P_2 := \frac{1}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y)\in (I_a\times I_b)-M_{a,b}} \underbrace{s_k(x,y)}_{\leq 1} = \frac{|I_a \times I_b - M_{a,b}|}{|I_a| \times |I_b| - |M_{a,b}|} = 1$$

Thus, Eq.(3) can be rewritten as

$$s_{k+1}(a,b) \leq \beta \times \big(\lambda + (1-\lambda)\big) + (1-\beta) = 1$$

On the other hand,

$$s_{k+1}(a,b) = \underbrace{\beta \times \big(\lambda \times P_1 + (1-\lambda) \times P_2\big)}_{\geq 0} + (1-\beta) \geq 1-\beta$$

3. **(Monotonicity)** We will prove by induction on k. For $k = 0$, $s_0(a,b) = 1$. According to Eq.(3), it follows that

$$\begin{aligned} s_1(a,b) &= \beta \times \left(\lambda \times \underbrace{\frac{\min\{|I_a|, |I_b|\}}{\max\{|I_a|, |I_b|\}}}_{\leq 1} + (1-\lambda) \times \underbrace{\frac{(|I_a| \times |I_b|) - |M_{a,b}|}{(|I_a| \times |I_b|) - |M_{a,b}|}}_{=1}\right) + (1-\beta) \\ &\leq \beta(\lambda + (1-\lambda)) + (1-\beta) = 1 = s_0(a,b) \end{aligned}$$

For $k > 0$, we assume that $s_{k+1}(a,b) \leq s_k(a,b)$ holds, and will prove that $s_{k+2}(a,b) \leq s_{k+1}(a,b)$ holds. According to Eq.(3), it follows that

$$\begin{aligned} s_{k+2}(a,b) &= \beta \times \left(\frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y)\in M_{a,b}} \overbrace{s_{k+1}(x,y)}^{\{\text{using hypothesis}\}\ \leq s_k(x,y)} \right. \\ &\qquad \left. + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y)\in (I_a\times I_b)-M_{a,b}} \underbrace{s_{k+1}(x,y)}_{\leq s_k(x,y)}\right) + (1-\beta) \\ &\leq \beta \times \left(\frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y)\in M_{a,b}} s_k(x,y) \right. \\ &\qquad \left. + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y)\in (I_a\times I_b)-M_{a,b}} s_k(x,y)\right) + (1-\beta) \\ &= s_{k+1}(a,b) \end{aligned}$$

□

A.2 Proof of Theorem 3

Proof. **(Existence)** For each pair of nodes (a, b), since the sequence $\{s_k(a, b)\}_k$ is lower-bounded by $(1 - \beta)$ (according to Property 2) and non-increasing (according to Property 3), by Monotone Convergence Theorem, $\{s_k(a, b)\}$ will converge to its infimum, denoted as $s(a, b)$, which is the exact RoleSim* solution, *i.e.,* $\lim_{k\to\infty} s_k(a, b) = s(a, b)$.

(Uniqueness) For each pair of nodes (a, b), suppose there exist two solutions, $s(a, b)$ and $\tilde{s}(a, b)$, that satisfy Eq.(3). We will prove that $s(a, b) = \tilde{s}(a, b)$. Let $\delta(a, b) := s(a, b) - \tilde{s}(a, b)$ and $\Delta := \max_{(a,b)}\{|\delta(a, b)|\}$. Then,

$$\begin{aligned}
\delta(a,b) &= s(a,b) - \tilde{s}(a,b) \\
&= \beta \times \Bigg(\frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y)\in M_{a,b}} \overbrace{s(x,y) - \tilde{s}(x,y)}^{=\delta(x,y)} \\
&\quad + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y)\in (I_a\times I_b) - M_{a,b}} \underbrace{s(x,y) - \tilde{s}(x,y)}_{=\delta(x,y)} \Bigg)
\end{aligned}$$

Therefore, taking the absolute value of both sides and applying triangle inequality $|x + y| \le |x| + |y|$ produces

$$\begin{aligned}
|\delta(a,b)| &\le \beta \times \Bigg(\frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \times \Bigg| \sum_{(x,y)\in M_{a,b}} \delta(x,y) \Bigg| \\
&\quad + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \times \Bigg| \sum_{(x,y)\in (I_a\times I_b) - M_{a,b}} \delta(x,y) \Bigg| \Bigg) \\
&\le \beta \times \Bigg(\frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \times \sum_{(x,y)\in M_{a,b}} \underbrace{\big|\delta(x,y)\big|}_{\le \Delta} \\
&\quad + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \times \sum_{(x,y)\in (I_a\times I_b) - M_{a,b}} \underbrace{\big|\delta(x,y)\big|}_{\le \Delta} \Bigg) \\
&\le \beta(\lambda \times \Delta + (1-\lambda) \times \Delta) = \beta \times \Delta \qquad (\forall a, b)
\end{aligned}$$

Thus, $\Delta = \max_{(a,b)}\{|\delta(a, b)|\} \le \beta \times \Delta$, implying $\Delta = 0$, *i.e.,* $s(a, b) = \tilde{s}(a, b)$. □

A.3 Proof of Theorem 4

Proof. We prove by induction on k. For $k = 0$, $s_0(a, b) = 1$. According to Property 2 of Theorem 1, $1 - \beta \le s_k(a, b) \le 1$, implying that $1 - \beta \le s(a, b) \le 1$. Thus, $s_0(a, b) - s(a, b) \le \beta$.

For $k > 0$, we assume that $s_k(a,b) - s(a,b) \leq \beta^{k+1}$ holds, and will prove that $s_{k+1}(a,b) - s(a,b) \leq \beta^{k+2}$ holds. Subtracting Eq.(3) from Eq.(1) produces

$$\begin{aligned} s_{k+1}(a,b) - s(a,b) = \beta \times \Bigg(& \frac{\lambda}{|I_a| + |I_b| - |M_{a,b}|} \sum_{(x,y)\in M_{a,b}} \overbrace{s_k(x,y) - s(x,y)}^{\leq \beta^{k+1}} \\ & + \frac{1-\lambda}{|I_a| \times |I_b| - |M_{a,b}|} \sum_{(x,y)\in (I_a \times I_b) - M_{a,b}} \underbrace{s_k(x,y) - s(x,y)}_{\leq \beta^{k+1}} \Bigg) \\ \leq \beta(\lambda \times \beta^{k+1} + (1-\lambda) \times \beta^{k+1}) = \beta^{k+2} & \qquad (\forall a,b) \qquad \square \end{aligned}$$

A.4 Proof of Lemma 1

Proof. Without loss of generality, we only consider the case of $|I_a| \leq |I_b| \leq |I_c|$. The proofs for other cases are similar, and omitted here due to space limitation. In this case, we have

$$|I_a| + |I_b| - |M_{a,b}| = \max\{|I_a|, |I_b|\} = |I_b|.$$

Hence, P_1 in Eq.(5) can be rewritten as

$$\begin{aligned} P_1 = & \frac{1}{|I_b|} \sum_{(x,y)\in M_{a,b}} s_k(x,y) + \frac{1}{|I_c|} \sum_{(y,z)\in M_{b,c}} s_k(y,z) - \frac{1}{|I_c|} \sum_{(x,z)\in M_{a,c}} s_k(x,z) \\ = & \overbrace{\left(\frac{1}{|I_b|} - \frac{1}{|I_c|}\right) \sum_{(x,y)\in M_{a,b}} s_k(x,y)}^{\text{Part 1(a)}} \\ & + \frac{1}{|I_c|} \underbrace{\left(\sum_{(x,y)\in M_{a,b}} s_k(x,y) + \sum_{(y,z)\in M_{b,c}} s_k(y,z) - \sum_{(x,z)\in M_{a,c}} s_k(x,z) \right)}_{\text{Part 1(b)}} \end{aligned} \tag{8}$$

To find an upper bound of Part 1(a), since $\sum_{(x,y)\in M_{a,b}} s_k(x,y) \leq \sum_{(x,y)\in M_{a,b}} 1 = |M_{a,b}|$, it follows that

$$\text{Part 1(a)} \leq \left(\frac{1}{|I_b|} - \frac{1}{|I_c|}\right) \times |M_{a,b}| = \left(\frac{1}{|I_b|} - \frac{1}{|I_c|}\right) \times |I_a| \tag{9}$$

To get an upper bound of Part 1(b), let

$$\begin{aligned} \tilde{I}_b &= \{y \mid \forall x \in I_a, \quad \exists y \in I_b, \quad s.t.\ (x,y) \in M_{a,b}\} \\ \tilde{M}_{a,c} &= \{(x,z) \mid \exists y \in I_b, \quad s.t.\ (x,y) \in M_{a,b} \ \wedge \ (y,z) \in M_{b,c}\} \end{aligned}$$

Then, $M_{b,c}$ can be partitioned into two parts: $M_{b,c} = M_{b,c}^{(1)} \cup M_{b,c}^{(2)}$ where

$$\begin{aligned} M_{b,c}^{(1)} &= \{(y,z) \in M_{b,c} \mid y \in \tilde{I}_b, \quad z \in I_c\} \\ M_{b,c}^{(2)} &= \{(y,z) \in M_{b,c} \mid y \in I_b - \tilde{I}_b, \quad z \in I_c\} \end{aligned}$$

Therefore,

$$\begin{aligned}
\text{Part 1(b)} &= \sum_{(x,y)\in M_{a,b}} s_k(x,y) + \sum_{(y,z)\in M_{b,c}} s_k(y,z) - \sum_{(x,z)\in M_{a,c}} s_k(x,z) \\
&= \Bigg(\sum_{(x,y)\in M_{a,b}} s_k(x,y) + \sum_{(y,z)\in M^{(1)}_{b,c}} s_k(y,z) \Bigg) + \sum_{(y,z)\in M^{(2)}_{b,c}} \underbrace{s_k(y,z)}_{\leq 1} - \sum_{(x,z)\in M_{a,c}} s_k(x,z) \\
&\leq \Bigg(\underbrace{\sum_{(x,z)\in \tilde{M}_{a,c}} s_k(x,z)}_{\leq \sum_{(x,z)\in M_{a,c}} s_k(x,z)} + |I_a| \Bigg) + \Big(\underbrace{|M_{b,c}|}_{=|I_b|} - \underbrace{|\tilde{I}_b|}_{=|I_a|} \Big) - \sum_{(x,z)\in M_{a,c}} s_k(x,z) \leq |I_b| \quad (10)
\end{aligned}$$

Substituting Eqs.(9) and (10) into (8) produces

$$P_1 \leq \left(\frac{1}{|I_b|} - \frac{1}{|I_c|} \right) |I_a| + \frac{|I_b|}{|I_c|} = \frac{|I_a|}{|I_b|} + \frac{|I_b| - |I_a|}{|I_c|} \leq \frac{|I_a|}{|I_b|} + \frac{|I_b| - |I_a|}{|I_b|} \leq 1 \qquad \square$$

For each $x \in I_a$, there exist $y_x \in I_b$ and $z_x \in I_c$ such that $(x, y_x) \in M_{a,b}$ and $(x, z_x) \in M_{a,c}$. Then, for each $z \in I_c - \{z_x\}$, there exists $y \in I_b$ such that

$$s_k(x,y) + s_k(y,z) - s_k(x,z) \leq 1$$

Summing both sides of the inequality over all $z \in I_c - \{z_x\}$ and all $y \in I_b$ yields

$$\underbrace{\sum_{y\in I_b} \sum_{z\in I_c - \{z_x\}} s_k(x,y)}_{\text{Part 2(a)}} + \underbrace{\sum_{y\in I_b} \sum_{z\in I_c - \{z_x\}} s_k(y,z)}_{\text{Part 2(b)}} - \underbrace{\sum_{y\in I_b} \sum_{z\in I_c - \{z_x\}} s_k(x,z)}_{=|I_b| \times \sum_{z\in I_c - \{z_x\}} s_k(x,z)} \leq (|I_c| - 1) \times |I_b|$$

where

$$\begin{aligned}
\text{Part 2(a)} &= (|I_c| - 1) \times \sum_{y\in I_b} s_k(x,y) \geq (|I_c| - 1) \times \sum_{y\in I_b - \{y_x\}} s_k(x,y) \\
\text{Part 2(b)} &= \sum_{(y,z)\in (I_b \times I_c)} s_k(y,z) - \underbrace{\sum_{y\in I_b} s_k(y,z_x)}_{\leq \sum_{(y,z)\in M_{b,c}} s_k(y,z)} \geq \sum_{(y,z)\in (I_b \times I_c) - M_{b,c}} s_k(y,z)
\end{aligned}$$

Therefore, it follows that

$$(|I_c| - 1) \times \sum_{y\in I_b - \{y_x\}} s_k(x,y) + \sum_{(y,z)\in (I_b \times I_c) - M_{b,c}} s_k(y,z) - |I_b| \times \sum_{z\in I_c - \{z_x\}} s_k(x,z) \leq (|I_c| - 1) \times |I_b|$$

Summing both sides of the inequality over all $x \in I_a$ produces

$$(|I_c|-1) \times \overbrace{\sum_{x\in I_a}\sum_{y\in I_b - \{y_x\}} s_k(x,y)}^{=\sum_{(x,y)\in (I_a\times I_b)-M_{a,b}} s_k(x,y)} + |I_a| \times \sum_{(y,z)\in (I_b\times I_c)-M_{b,c}} s_k(y,z) - |I_b| \times \underbrace{\sum_{x\in I_a}\sum_{z\in I_c-\{z_x\}} s_k(x,z)}_{=\sum_{(x,z)\in (I_a\times I_c)-M_{a,c}} s_k(x,z)} \leq |I_a| \times (|I_c|-1) \times |I_b|$$

Since $\bigcup_{x\in I_a} \{(x,y_x)\} = M_{a,b}$ and $\bigcup_{x\in I_a} \{(x,z_x)\} = M_{a,c}$, we divide both sides of the inequality by $(|I_a| \times (|I_c|-1) \times |I_b|)$ to get $P_2 \leq 1$ in Eq.(6). □

A.5 Proof of Theorem 2

Proof. By the definition of $d(a,b) := 1 - s(a,b)$, based on the fact that

$$\begin{aligned} d(a,b) + d(b,c) \geq d(a,c) \quad &\Leftrightarrow \quad 1 - s(a,b) + 1 - s(b,c) \geq 1 - s(a,c) \\ &\Leftrightarrow \quad s(a,b) + s(b,c) - s(a,c) \leq 1 \end{aligned} \tag{11}$$

in what follows we will prove Eq.(11) holds by induction on k. For $k = 0$, by virtue of Eq.(2), it is apparently that

$$s_0(a,b) + s_0(b,c) - s_0(a,c) = 1 + 1 - 1 = 1 \leq 1.$$

For $k > 0$, we assume that $s_k(a,b) + s_k(b,c) - s_k(a,c) \leq 1$ holds, and will prove that $s_{k+1}(a,b) + s_{k+1}(b,c) - s_{k+1}(a,c) \leq 1$ holds.

Let P_1 and P_2 be defined by Eqs.(5) and (6), respectively. According to Lemma 1, it follows from $P_1 \leq 1$ and $P_2 \leq 1$ that

$$\begin{aligned} s_{k+1}(a,b) + s_{k+1}(b,c) - s_{k+1}(a,c) &= \beta(\lambda \times P_1 + (1-\lambda) \times P_2) + (1-\beta) \\ &\leq \beta(\lambda \times 1 + (1-\lambda) \times 1) + (1-\beta) \leq 1. \quad \square \end{aligned}$$

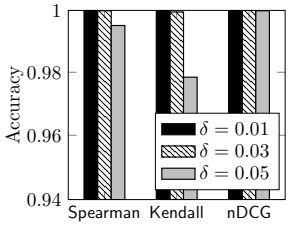
B Description of RoleSim* Algorithm

Algorithm. The fixed-Point scheme in Eqs.(2) and (3) implies an iterative algorithm for RoleSim* computation, as illustrated in Algorithm 1. It starts initialising all pairs of similarities to 1s (line 1), and carries out iterative computations of similarities for each pair of nodes (lines 3–15). If there are no in-neighbors for node a or b, $s(a,b)$ is set to $1-\beta$ (lines 4–6). Otherwise, it finds maximum weighed matching $M_{a,b}$ in bipartite graph $(I_a \cup I_b, I_a \times I_b)$ (8), and averages the $(k-1)$-th iterative similarities over $M_{a,b}$ (*resp.* $(I_a \times I_b) - M_{a,b}$) to get w_1 (*resp.* w_2) (lines 9–14). Then, the weighted average of w_1 and w_2 is returned as score $s_k(a,b)$ at k-th iteration. This process continues till all pairs of similarities are computed for each iteration.

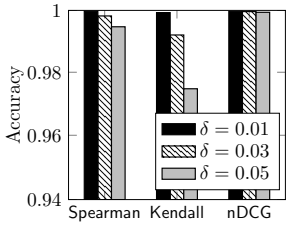
Algorithm 1: RoleSim* (G, β, λ, K)

Input : digraph $G = (V, E)$, decay factor β, relative weight λ, #-iterations K
Output: RoleSim* scores $s_K(*, *)$.

initialise $s_0(*, *) := 1$
for $k := 1, 2, \cdots, K$ **do**
  foreach *pair* $(a, b) \in V^2$ **do**
    $I_a := \{x \in V \mid (x, a) \in E\}$, $I_b := \{x \in V \mid (x, b) \in E\}$
    if $I_a = \emptyset$ *or* $I_b = \emptyset$ **then**
      $s_k(a, b) := 1 - \beta$
    else
      $M_{a,b} :=$ maximum matching in bipartite graph $(I_a \cup I_b, I_a \times I_b)$
      initialise $t_1 := 0$ and $t_2 := 0$
      foreach $(x, y) \in M_{a,b}$ **do**
        $t_1 := t_1 + s_{k-1}(x, y)$
      foreach $(x, y) \in (I_a \times I_b) - M_{a,b}$ **do**
        $t_2 := t_2 + s_{k-1}(x, y)$
      $w_1 := \lambda / (|I_a| + |I_b| - |M_{a,b}|)$, $w_2 := (1 - \lambda) / (|I_a| \times |I_b| - |M_{a,b}|)$
      $s_k(a, b) := \beta \times (w_1 \times t_1 + w_2 \times t_2) + (1 - \beta)$
return $s_K(*, *)$



(a) Varying δ on DBLP



(b) Varying δ on AMZ

	k	$\delta = 0.01$	$\delta = 0.03$	$\delta = 0.05$
DBLP	1	0	0	0
	2	0	0	0
	3	94,192	154,058	848,639
	4	6,823	1,622,334	4,381,770
	k	$\delta = 0.01$	$\delta = 0.03$	$\delta = 0.05$
AMZ	1	0	0	0
	2	0	0	1,958
	3	6,913,099	7,172,106	8,109,031
	4	2,199,304	4,503,986	5,117,371

(c) # of Pruned Pairs

Fig. 5: Accuracy Comparison for Different Threshold-Based RS*

C Additional Experiments

Effect of Threshold δ on RS* Accuracy. Figures 5(a) and 5(b) show the influence of threshold δ on RS* accuracy over real datasets (DBLP and AMZ). The accuracy is evaluated using three ranking measures (Spearman, Kendall, nDCG). We randomly sample 100 queries from each dataset, and vary threshold δ from 0.01 to 0.05. For each δ, we compute single-source threshold based RS* similarities $\{s_k^\delta(*, q)\}$ *w.r.t.* each query q. Choosing non-threshold based RS* similarities $\{s_k(*, q)\}$ as the baseline, we evaluate the average value of Spearman, Kendall, and nDCG, respectively, for each threshold based RS* over 100 queries on each dataset. We notice that, on each dataset, all the threshold based RS* consistently achieve $> 98\%$ accuracy by each ranking measure. For top-100 results on both datasets, the similarity rankings attain $> 99\%$ nDCGs on average. These imply that the accuracy compromised by the threshold based RS*

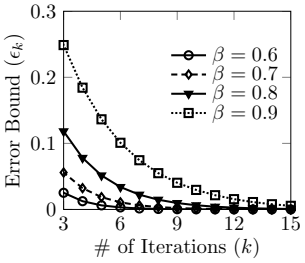


Fig. 6: Effect of (k, β) on ϵ_k

	$\beta = 0.6$		$\beta = 0.8$			$\beta = 0.6$		$\beta = 0.8$	
#-Iter. (k)	**Actual Error** (ϵ_k)	**Est. Bound** (β^{k+1})	**Actual Error** (ϵ_k)	**Est. Bound** (β^{k+1})	**#-Iter.** (k)	**Actual Error** (ϵ_k)	**Est. Bound** (β^{k+1})	**Actual Error** (ϵ_k)	**Est. Bound** (β^{k+1})
1	0.1029	0.3600	0.2717	0.6400	6	0.0031	0.0280	0.0339	0.2097
2	0.0509	0.2160	0.1793	0.5120	7	0.0015	0.0168	0.0224	0.1678
3	0.0252	0.1296	0.1183	0.4096	8	0.0008	0.0101	0.0147	0.1342
4	0.0125	0.0778	0.0780	0.3277	9	0.0004	0.0060	0.0097	0.1074
5	0.0062	0.0467	0.0515	0.2621	10	0.0002	0.0036	0.0064	0.0859

Table 3: Actual & Estimate Error for $\beta \in \{0.6, 0.8\}$

is negligibly small for fast speed. Moreover, when δ increases from 0.01 to 0.05, the accuracy decreases slightly for each ranking measure because large threshold may prune a large number of node-pairs per iteration. This agrees well with the pruning table in Figure 5(c), where large δ implies more pairs are eliminated at each iteration.

Iterative Error. Finally, we evaluate the effects of number of iterations k on the iterative error of RS*. The error is measured by difference ϵ_k between k-th iterative score $s_k(*,*)$ and exact one $s(*,*)$. We only report the results for a pair of nodes on DBLP since the trends for other pairs and on other datasets (AMZ and SYN) are similar and omitted here due to space limitations. For each pair of nodes on DBLP, we fix damping factor β, and vary k from 1 to 15.

Figure 6 depicts how k-th iterative error ϵ_k changes with k. It is discerned that, for any given damping factor β, ϵ_k exponentially decreases to 0 as k grows. The larger damping factor β will cause a shift outward in the accuracy curve, thereby exhibiting the slower convergence rate of RoleSim* iterations. In addition, at each iteration k, it is noticed that small settings of damping factor β will lead to small iterative error of RoleSim*. These agree well with our theoretical bound $k = \lceil \log_\beta \epsilon_k \rceil$ in Theorem 3 for RoleSim* accuracy analysis.

The actual and the estimated error bound value for $\beta = 0.6$ and $\beta = 0.8$ per 10 iteration are illustrated in Table 3, which shows for each iteration the computed actual error bounds are completely compatible with the theoretical estimated error bounds.