Scalable and Axiomatic Ranking of Network Role Similarity

RUOMING JIN, Kent State University
VICTOR E. LEE, John Carroll University
LONGJIE LI, Lanzhou University

A key task in analyzing social networks and other complex networks is role analysis: describing and categorizing nodes according to how they interact with other nodes. Two nodes have the same role if they interact with *equivalent* sets of neighbors. The most fundamental role equivalence is automorphic equivalence. Unfortunately, the fastest algorithms known for graph automorphism are nonpolynomial. Moreover, since exact equivalence is rare, a more meaningful task is measuring the role *similarity* between any two nodes. This task is closely related to the structural or link-based similarity problem that SimRank addresses. However, SimRank and other existing similarity measures are not sufficient because they do not guarantee to recognize automorphically or structurally equivalent nodes. This article makes two contributions. First, we present and justify several axiomatic properties necessary for a role similarity measure or metric. Second, we present RoleSim, a new similarity metric that satisfies these axioms and can be computed with a simple iterative algorithm. We rigorously prove that RoleSim satisfies all of these axiomatic properties. We also introduce Iceberg RoleSim, a scalable algorithm that discovers all pairs with RoleSim scores above a user-defined threshold θ. We demonstrate the interpretative power of RoleSim on both both synthetic and real datasets.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; E.1 [**Data Structures**]: Graphs and Networks

General Terms: Theory, Algorithms, Measurement, Experimentation

Additional Key Words and Phrases: Complex network, social network, vertex similarity, role similarity, ranking, automorphic equivalence

ACM Reference Format:
Ruoming Jin, Victor E. Lee, and Longjie Li. 2014. Scalable and axiomatic ranking of network role similarity. ACM Trans. Knowl. Discov. Data 8, 1, Article 3 (February 2014), 37 pages.
DOI: http://dx.doi.org/10.1145/2518176

1. INTRODUCTION

In social science, it is well established that individual agents tend to play roles or assume positions within their interaction network. For instance, in a university, each individual can be classified into the position of faculty member, administration, staff, or student. Indeed, role discovery is a major research subject in classical social science [Wasserman and Faust 1994]. Interestingly, recent studies have found that roles not only appear in other types of networks, including food webs [Luczkovich et al. 2003], world trade [Hafner-Burton et al. 2009], and even software systems [Dragan et al.

Longjie Li is a Visiting Scholar at Kent State University.
Authors' address: R. Jin (corresponding author), Department of Computer Science, Kent State University, Kent, OH; email: jin@cs.kent.edu.

DOI: http://dx.doi.org/10.1145/2518176

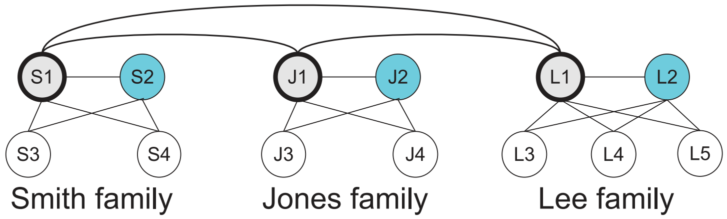


Fig. 1. Example graph for role equivalence.

2009], but also can help to predict node functionality within their domains. For example, in a protein interaction network, proteins with similar roles tend to serve similar metabolic functions. Thus, if we know the function of one protein, we can predict that all other proteins having a similar role would also have similar function [Holme and Huss 2005]. In other cases, such as online social networks, there are no a priori role categories. The classification must be learned based on the interaction patterns.

Role discovery is complementary to graph clustering, an important tool for analyzing network structures. Graph clustering attempts to decompose a graph into densely connected components. It produces a high-level structural model consisting of a small number of "cluster nodes" and the "super edges" between these cluster nodes. The clustering scheme inevitably overlooks and oversimplifies the interaction patterns of the individual nodes. In reality, the nodes within a cluster may take very different roles: some of them may serve as the core of the clusters, some may be peripheral nodes, and some serve as the connectors to link between clusters. At the same time, nodes located in different clusters might play similar roles. Furthermore, even when a network lacks modularity structure, such as in a simple hierarchical structure, roles can still characterize the interaction patterns of each node. Hence, roles provide an orthogonal abstraction for simplifying and highlighting the complex interactions among nodes.

A central question in studying the roles in a network system is how to define *role similarity*. In particular, how can we rank two nodes' role similarity in terms of their interaction patterns? Despite its vital importance for network analysis and decades of work by social scientists, joined recently by computer scientists, no satisfactory metric for role similarity has yet emerged. A key issue is the encapsulation of graph automorphism into a role similarity metric: *if two nodes are automorphically equivalent, then they should share the same role and their role similarity should be maximal*. From a network topology viewpoint, automorphic nodes have equivalent surroundings, so one can replace the other. Figure 1 illustrates a graph with nodes $S1$ and $J1$ being automorphically equivalent.

The traditional social science approach for role analysis has been to define suitable mathematical equivalence relations for nodes so that they can be partitioned into equivalence classes (roles). An essential property of these equivalences is that they should positively confirm automorphic equivalence—that is, if any two nodes are automorphic, then they are role equivalent. (The converse is not necessarily true.) Confirming automorphism is an instance of verifying a solution, which is often algorithmically less complex than discovering a solution. Thus, although there is no known polynomial-time algorithm for discovering graph automorphism,[1] role equivalence algorithms [Batagelj et al. 1992; Borgatti and Everett 1993; Sparrow 1993] can still guarantee to satisfy the

[1] The computational complexity of graph isomorphism and automorphism are still unproven to be either P or $NP - Complete$.

aforementioned automorphism confirmation property. These equivalence rules also directly correspond to the aforementioned coloration.

However, by relying on strict equivalence rules, these role modeling schemes can produce only binary similarity metrics: two nodes are either equivalent (similarity = 1) or not (similarity = 0). In real-world networks, usually only a very small portion of the node-pairs would satisfy an equivalence criteria [MacArthur et al. 2008], and among those, many are simply trivially equivalent (such as singletons or children of the same parent). In addition, strict rule-based equivalence is not robust with respect to network noise, such as false-positive or false-negative interactions. Thus, it is desirable in many real-world applications to rank node-pairs by their degree of similarity or provide a real-valued node similarity *metric*.

Recent research works have proposed various measures of node similarity based on similarity of interactions. In Leicht et al. [2005], a accrues similarity to b if a has a neighbor that is similar to b. This assumes that neighbors (a and its neighbors) should be somewhat similar; however, roles should be class-based, not proximity based. Jeh's SimRank [Jeh and Widom 2002] is based on the following principle: "two nodes are similar if they link to similar nodes." Mathematically, for any two different nodes x and y, SimRank computes their similarity recursively according to the average similarity of all neighbor-pairs (a neighbor of x paired with a neighbor of y). A single node has self-similarity value 1. This is equivalent to the probability that two simultaneous random walkers, starting at x and y, will eventually meet. Most other node structural similarity measures [Antonellis et al. 2008; Fogaras and Rácz 2005; Li et al. 2009; Xi et al. 2005; Yin et al. 2006; Zhao et al. 2009] are variants of SimRank. Although SimRank seems to capture the intuition of the presented recursive structural similarity, its random walk matching does not satisfy the basic graph automorphism condition. For example, in Figure 1, although $S1$ and $J1$ are automorphically equivalent, SimRank assigns them a value of 0.226. We discuss this further in Section 3.2. To our best knowledge, there is no available real-valued structural similarity measure satisfying the automorphic equivalence requirement. Since automorphic equivalence is a pivotal characteristic of the notion of role, its lack disqualifies these existing measures from serving as authentic role similarity measures.

Thus, we have an open problem: *can we derive a real-valued role similarity measure or ranking that complies with the automorphic equivalence requirement?* In this article, we develop the first real-valued similarity measure to solve this problem. In addition, our measure is also a metric—that is, it satisfies the triangle inequality. The key feature of our role similarity measure is a weighted generalization of the *Jaccard coefficient* to measure the neighborhood similarity between two nodes. Unlike SimRank, which considers the average similarity among all possible pairings of neighbors, our measure considers only those pairs in the optimal matching of their two neighbor sets that maximizes the targeted similarity function.

The article is organized as follows: Section 2 provides a detailed review of the existing works on node similarity. Section 3 presents axiomatic properties of any real-valued similarity measure, including a requirement for automorphic equivalence. We also show that SimRank does not satisfy the automorphic equivalence requirement. Section 4 describes our RoleSim measure, its computation, and its correctness with respect to the axiomatic properties. Section 5 presents Iceberg RoleSim, a scalable algorithm that discovers all pairs with RoleSim scores above a user-defined threshold. Section 6 provides experimental validation and evaluation of our proposed approach for ranking the the role similarity between vertex pairs.

An earlier version of the article [Jin et al. 2011] is published in KDD'11. This article significantly extends that work and makes the following additional contributions: (1) we provide a detailed and extensive survey of the existing node similarity work (Section 2);

(2) we introduce a new Iceberg RoleSim to scale the existing RoleSim computation (Section 5); (3) we perform a detailed evaluation of Iceberg RoleSim to demonstrate its performance and scalability (Section 6.5); and (4) we present a detailed case study comparing RoleSim with the state-of-the-art approaches using a coauthor network (Section 6.6).

2. RELATED WORK

This section provides a survey of prior work relevant to our core problem of role similarity. The first section describes several formal definitions of role and role equivalence. The next section reviews existing work on role similarity. The remaining sections review numerous measures for the general problem of local structural similarity, considering first centrality-based measures and then link-based measures.

We take this opportunity to establish some symbolic notation to use here and in the remainder of this work. We use the terms *graph* and *network* interchangeably; the same is true for *vertex* and *node*. In most instances, we speak of networks and nodes, as this is the more common usage in the principal application domains of interest, but we revert to graph and vertex at times when speaking in a graph-theoretical sense.

We define a graph or network $G = (V, E)$ as a set of nodes V and a set of connecting edges $E \subseteq V \times V$. The neighbors of a node v are those nodes that are joined directly to v with an edge. The set of neighbors $N(v) = \{u|(u, v) \in E\}$. The degree of v is $d_v = |N(v)|$. When discussing computational complexity, the number of vertices in a graph is $n = |V|$, and the number of edges is $m = |E|$. By default, we assume that edges are undirected, but all of the concepts and formulas in this work can be extended to directed graphs by computing scores using in-neighbors and out-neighbors separately and then combining the scores.

2.1. Role Equivalence

Computing role similarity encompasses two more fundamental problems: what is a role, and how should we measure closeness to role? We use the following definition of role:

Definition 1 (*Role and Role Equivalence*). A *role* is the set of relationships between an individual and others. In graph theory terms, the role of v is the set of all edges incident to v. For an undirected graph: *role*$(v) = \{(u, v) \in E\}$. Two individuals fulfill *equivalent roles* if they have equivalent relationships.

For example, consider Figure 1, which depicts three siblings $\{S1, J1, L1\}$, who are each a parent in a family. Each family has two parents and either two or three children. There are three types of relationships shown:

(1) Spouse {S1-S2, J1-J2, L1-L2}
(2) Parent–Child {S1-S3, S1-S4, S2-S3, S2-S4, J1-J3, etc.}
(3) Sibling {S1-J1, S1-L1, J1-L1}

For simplicity, we do not show the sibling relationships in the younger generation.

Intuitively, S1 and J1 appear to be role equivalent: each is a spouse, a parent of two children, and the sibling of two others. Note we have not labeled or colored the edges, only the nodes. For example, a parent-child relationship is defined by the two participating nodes, not by a prelabeling of the edge. However, we do not know that the two ends represent a parent and a child until we identify the roles. In general, even the nodes will not be labeled or colored in advance. We will begin only with a graph topology; the role equivalence discovery problem is to identify the colorings.

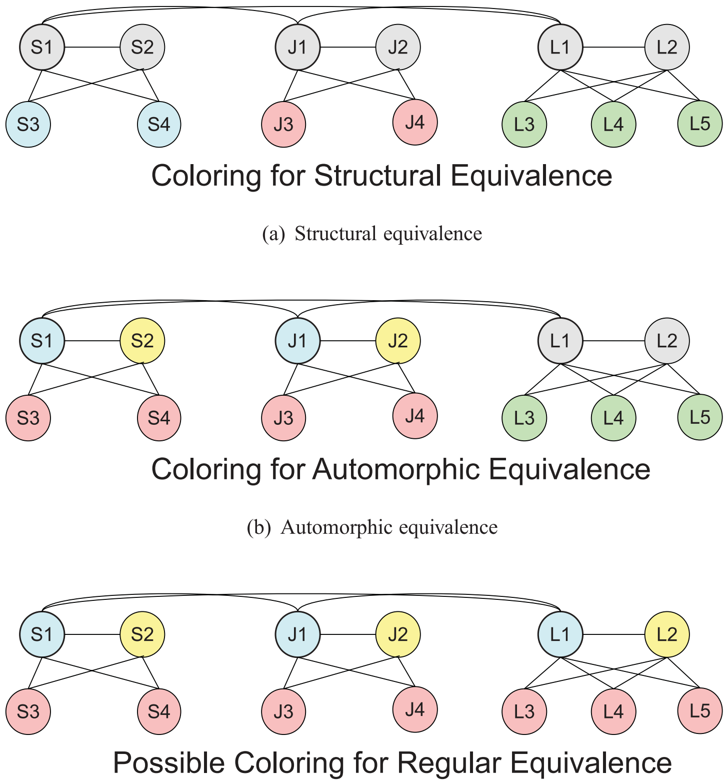
In social network analysis, the traditional approach for discovering role groups is to define a equivalence relation and to partition the actors into equivalence classes.

Table I. Equivalence Classes for Figure 1

Equivalence	Neighbor Rule	Nonsingleton Classes	Unique Partitioning?
Structural	Same nodes ($N(u) = N(v)$)	{S3,S4}, {J3,J4}, {L3,L4,L5}	Yes
Automorphic	For automorphism σ, $\forall x \in N(u), \exists y \in N(v)$ s.t. $y = \sigma(x)$	{S1,J1}, {S2,J2}, {S3,S4,J3,J4}, {L3,L4,L5}	Yes
Equitable partition	Same number per class	{S1,J1}, {S2,J2}, {S3,S4,J3,J4}, {L3,L4,L5}	No
Regular	Same classes	{S1,J1,L1}, {S2,J2,L2}, {S3,S4,J3,J4,L3,L4,L5}	No

Actors who fulfill the same role are equivalent. Over the years, four definitions have stood out. These four, in decreasing order of strictness, are structural equivalence, automorphic equivalence, equitable partition, and regular equivalence. Table I shows how these different definitions generate different roles from the same network.

- **Structural Equivalence:** Two actors are *structurally equivalent* if they interact with the *same* set of others [Lorrain and White 1971]. Mathematically, u and v are structurally equivalent if and only if $N(u) = N(v)$. For example, consider the extended family shown in Figure 1. $S1$, $J1$, and $L1$ are siblings; $S2$, $J2$, and $L2$ are spouses; and the remaining nodes are their children. Each family's children, $\{S3, S4\}$, $\{J3, J4\}$, and $\{L3, L4, L5\}$, form a nontrivial equivalence class. However, none of the parents can be grouped together via structural equivalence. Figure 2(a) illustrates this partitioning. Nodes with the same color are in the same class, except gray nodes represent singleton classes. Each gray node is its own class. This model is too strict to be useful for simplifying a large network and to discover meaningful roles.
- **Automorphic Equivalence:** Two actors (nodes) u and v are *automorphically equivalent* if there is an automorphism σ of G where $v = \sigma(u)$ [Borgatti and Everett 1992]. An automorphism σ of a graph G is a permutation of vertex set V such that for any two nodes u and v, $(u, v) \in E$ iff $(\sigma(u), \sigma(v)) \in E$. In social terms, u and v can swap names, along with possibly some other name swaps, while preserving all of the actor-actor relationships. Let $\Gamma(G)$ be the group of all automorphisms of graph G. For any two nodes u and v in G, $u \equiv v$ if $u = \sigma(v)$ for some $\sigma \in \Gamma(G)$. Note that $\equiv$ is an equivalence relation on V; if $u \equiv v$, we say that u is automorphically equivalent to v. The equivalence classes generated under $\Gamma(G)$ (or $\equiv$) are called *orbits*. The equivalence class for vertex $v \in V$ is called the orbit of v and denoted as $\Delta(v) = \{\sigma(v) \in V, \sigma \in \Gamma(G)\} = \{u \mid u \equiv v\}$. Each orbit corresponds to a role in the automorphic equivalence. Understanding the importance of automorphic equivalence and applying it to role modeling was a major breakthrough in classical social network research. In our example Figure 1, from the topology alone, we cannot distinguish between the Smith family and the Jones family. The Lee family is distinct because it has three children instead of two. Therefore, the equivalence classes are $\{S1, J1\}$, $\{S2, J2\}$, $\{S3, S4, J3, J4\}$, $\{L1\}$, $\{L2\}$, and $\{L3, L4, L5\}$ (Figure 2(b)). Interestingly, automorphically equivalent classes must have equivalent indirect relations as well, such as equivalent in-laws and cousins. However, automorphic equivalence is hard to compute and still very strict.
- **Exact Coloration (Equitable Partition):** An *exact coloration* of graph G assigns a color to each node, such that any two nodes share the same color if and only if they have the same number of neighbors of each color [Everett and Borgatti 1996]. Nodes of the same color form an equivalence class. An exact coloration is also referred to as equitable partition [Godsil and Royle 2001] and graph divisor [Cvetkovíc et al. 1998] and is often applied in the vertex classification/refinement for canonical labeling in a graph isomorphism test [Read and Corneil 1977; McKay 1981]. A graph may


S1
S2
J1
J2
L1
L2
S3
S4
J3
J4
L3
L4
L5
Coloring for Structural Equivalence
Coloring for Automorphic Equivalence
Possible Coloring for Regular Equivalence

(a) Structural equivalence

(b) Automorphic equivalence

(c) Regular equivalence

Fig. 2. Comparing equivalence schemes (gray nodes are not equivalent).

have several exact colorations; in general, we seek the fewest colors. In our example, structural equivalence and automorphic equivalence offer two different exact colorations. Exact coloration relaxes automorphism by considering only immediate neighborhood equivalence, yet it still embodies a recursive aspect to role modeling.

- **Regular Equivalence (Bisimulation):** Two actors are *regularly equivalent* if they interact with the same variety of role classes, where class is recursively defined by regular equivalence [White and Reitz 1983]. Unlike automorphic equivalence and exact coloration, regular equivalence does not care about the cardinality of neighbor relationships, only whether they are nonzero. For example, using regular equivalence, all three families could be equivalent, with only three equivalence classes: *sibling–parent*$\{S1, J1, L1\}$, *spouse–parent*$\{S2, J2, L2\}$, and *child* (Figure 2(c)). Note that under regular equivalence, any two automorphically equivalent nodes may be merged into the same regular equivalence class. In computer science, the regular equivalence is often referred to as the bisimulation, which is widely used in automata and modal logic [Marx and Masuch 2003].

2.2. Existing Role Similarity Measures

We now move from strict equivalence to measuring similarity. There has been limited work on measuring role similarity. For structural equivalence, one can count how many neighbors they share, normalized by some factor. However, we noted in the previous section that structural similarity is too limiting for our interest.

Two algorithms for measuring the extent of regular equivalence are described in Borgatti and Everett [1993]. However, the authors acknowledge “the lack of a theoretical rationale for the measure of similarity produced.” The core of the problem lies not in their algorithms but in regular equivalence itself. Both regular equivalence and exact coloration are problematic because there may be more than one equivalence partitioning for a given graph. Indeed, for regular equivalence, every graph has two degenerate partitionings: (1) place all nodes in one class and (2) place each node in its own class (except structurally equivalence nodes may be in the same class). If one is measuring similarity, from which partition are you measuring the similarity?

To find the “best” regular partitioning, one can consider an information-theoretic or Minimum Description Length (MDL) approach: group nodes into classes or blocks that approximately describe a true regular equivalence class membership. This is the blockmodeling approach [Batagelj et al. 1992; Doreian et al. 2005]. MDL blockmodeling tries to solve the following optimization problem: assign n nodes to b blocks such that the aggregate cost of describing the block structure $O(b \log b)$ plus the cost of describing the difference between the appropriate block structure and the true structure is minimized. Heuristically, the problem is easier if the number of blocks b is preset, but whether it is or not, the exact optimization problem is NP hard. Furthermore, it does not truly address our problem: we want to know the similarity of nodes at the individual level. Blockmodeling jumps ahead to a global partitioning problem and only provides a rough measure of distance.

2.3. Structural Similarity Measures

Due to the limited work in role similarity, we look at prior work for other types of structural similarity, namely (1) centrality of a node with respect to the full graph and (2) link-based similarity. We will not consider density measures. Density has been well studied in other works [Lee et al. 2010] and is not relevant to our definition of role similarity.

2.3.1. Similarity of Node Centrality. This section discusses properties of individual nodes in the context of a network. By themselves, these properties are not similarity measures. However, the property values of two different nodes can be compared to produce a similarity measure. Node degree, closeness centrality, and betweenness centrality are three such measures. Degree counts the number of incident edges. In directed graphs, it can be divided into in-degree and out-degree. *Closeness centrality* is the average distance between a node v and every other node in the graph. *Betweenness centrality* measures how frequently a node lies on the path between two other nodes [Freeman 1977]. These definitions, however, are too limited to encompass the concept of role. Role is not merely centrality of degree, closeness, or betweenness. It is any or all of these and possibly more; it is whatever makes the structural position of a node unique.

2.3.2. Link Similarity. Another way that node structural similarity has been defined is in terms of link similarity—that is, how are two nodes connected to one another? One of the earliest measures of link similarity is *bibliographical coupling* [Kessler 1963]. This measures the similarity between two research publications by counting the number of works that are listed in both of their bibliographies. *Co-citation* [Small 1973] turns this around by counting the number of later works that cite both of the two original documents. As the size of a work's bibliography increases, the likelihood that it will contain a particular work increases. Therefore, a common normalization of these two measures is to divide the count by the number of distinct works cited.

We can form a *citation graph*, where each node is a document and a directed edge (a, b) means that document a cites document b. Let $I(a)$ and $O(a)$ be the in-neighbor set and out-neighbor set of a, respectively. Let I_a and O_b be the in-degree and out-degree of a. Then, the normalized bibliographic coupling index is

$$S_{bc}(a,b) = \frac{|O(a) \cap O(b)|}{|O(a) \cup O(b)|}, \tag{1}$$

and the normalized co-citation index is

$$S_{cc}(a,b) = \frac{|I(a) \cap I(b)|}{|I(a) \cup I(b)|}. \tag{2}$$

These are simply the Jaccard coefficients of the out-neighbor sets and in-neighbors sets, respectively.

These two are suitable for unweighted and directed graphs. If a graph is undirected, then the two measures are the same. Suppose that we have a weighted graph, though. This could be an author-collaboration graph, where edge (a, b) counts how many times author a has worked with author b. Or, it could be a bipartite document-term graph, where edge (d_a, t_b) counts the number of times that document a uses term b. Assign to each node a feature vector. For a coauthorship graph, each author is a feature dimension; its feature vector is the set of edge weights to every other author. For a document-term bipartite graph, a document has a term vector, weighted according to term frequencies of the document. If we represent the graph as an adjacency matrix, then the feature vector of node i is the i_{th} row of the matrix.

Given this representation, the cosine between two objects is a convenient and meaningful measure. Identical documents have cosine of 1, and documents with no features in common are orthogonal with cosine of 0.

$$S_{cit}(a,b) = \frac{A \cdot B}{\|A\| \|B\|}, \tag{3}$$

where A is the feature vector of node a.

A small modification to the denominator of Equation (3), attributed to Tanimoto [1958], maintains the overall behavior of the similarity function while aligning it with the Jaccard coefficient when the feature vectors are binary valued:

$$S_{tani}(a,b) = \frac{A \cdot B}{||A||^2 + ||B||^2 - A \cdot B}, \tag{4}$$

Schultz and Liberman [1999] adapted the well-known TF-IDF query-document similarity measure to produce a term-weighted document-document similarity measure. Here, $A(t)$ is the frequency of term t for object a, and $idf(t)$ is the inverse document frequency for term t. More generally, it is the significance or importance of term t appearing in a document.

$$S_{wcos}(a,b) = \frac{\sum_{t \in T} A(t)B(t)idf(t)}{||A||\ ||B||} \tag{5}$$

SimRank. Jeh and Widom [2002] realized that a more general way to attack the node similarity problem was to not only look for shared neighbors—that is, neighbors that are *identical*—but to look for neighbors that are *similar*. This produces the following recursive statement: “two objects are similar if they are related to similar objects” [Jeh and Widom 2002]. Formally, their SimRank measure is defined as follows:

$$sim_{sr}(a,b) = \frac{c}{|I(a)|\ |I(b)|} \sum_{x \in I(a)} \sum_{y \in I(b)} sim_{sr}(x,y) \tag{6}$$

if $a \neq b$. If $a = b$, then $sim_{sr}(a,b) = 1$. c is a constant $0 < c < 1$. In addition, for SimRank and all of its variants, if either a or b has no neighbors, then $sim(a,b) = 0$. SimRank can be computed iteratively by initializing the matrix of $sim(.)$ values, hereafter called the S matrix, to the identity matrix. A SimRank similarity value can be interpreted as the probability that two simultaneous random walkers, starting at a and b, will eventually meet if they traverse the network backward along directed edges. At each step, there is a probability $(1-c)$ that one or both walkers will abandon the walk and exit the network.

Obviously, we can add the effects of in-neighbors and out-neighbors to produce a more comprehensive measure of the neighbor similarity between two objects. Several authors have proposed this [Lin et al. 2007; Zhao et al. 2009].

SimRank can be described as a recursive extension of the co-citation index. An important difference between the noniterative algorithms in Section 2.3.2 and SimRank is that the earlier algorithms can be computed locally with a minimum of computational effort. With SimRank, however, to compute the similarity of even a single pair of objects, one has to consider the entire graph. This increases the computational requirements by a factor of n^2k, where k is the number of iterations. Consequently, several authors [Lizorkin et al. 2008; Jia et al. 2009; Cai et al. 2009; Li et al. 2009] have worked to reduce both the computational and memory requirements for SimRank, for general and specific applications.

In addition to concerns about the computational efficiency of the original SimRank formula, there are some structural flaws that mar its elegance. First, SimRank scores sometimes decrease when we would intuitively expect them to increase. Suppose that we have an object-pair that has all neighbors in common. Then $sim_{sr}(a, b) = c/d$, where d is the degree of a or b. As d increases, this should mean stronger ties between a and b, but clearly sim_{sr} actually decreases.

SimRank++. Antonellis et al. [2008] partially compensates for this unwanted decrease by inserting an *evidence* factor. The more neighbors in common, the higher the evidence of similarity. They define evidence as

$$ev(a, b) = \sum_{i=1}^{|N(a)\cap N(b)|} \frac{1}{2^i}, \tag{7}$$

where $N(a)$ is the undirected neighbor set of a. If a and b have only one neighbor in common, $ev = 1/2$. As the number of neighbors increases, $ev \rightarrow 1$. This yields the following similarity definition:

$$sim_{ev}(a, b) = ev(a, b) \cdot c \sum_{x=1}^{N(a)} \sum_{y=1}^{N(b)} sim_{ev}(x, y). \tag{8}$$

The very narrow range [0.5, 1] of the evidence factor, however, leads to the problem that $sim_{ev}(.)$ values are no longer bounded to a maximum of 1 or even to a constant. Instead, the maximum depends on the maximum value of $||N(a)|| \cdot ||N(b)||$ for the graph. The authors make one more extension to support edge-weighted graphs. Their final measure is called SimRank++:

$$sim_{spp}(a, b) = ev(a, b) \cdot c \sum_{x=1}^{N(a)} \sum_{y=1}^{N(b)} w_{ab} w_{by} sim_{spp}(x, y). \tag{9}$$

PSimRank. Fogaras and Rácz [2005] realize that the cause of improper weighting of neighbor matching in SimRank is due to the paired-random walk model. Ignoring the decay constant c for the moment, SimRank values are equal to the probability that two simultaneous random walkers, starting at nodes a and b, will eventually encounter each other. Even in the best-case scenario, in which a and b have all of the same neighbors in common, so that $N(a) = N(b)$, the probability that the two walkers will happen to choose the same neighbor is $1/d_a$, which decreases as the degree increases. To amend this situation, Fogaras and Rácz introduce coupled random walks. They partition the event space into three cases:

(1) Probability $P_1 = P(a \text{ and } b \text{ step to the same node}) = \frac{|I(a) \cap I(b)|}{|I(a) \cup I(b)|}$
(2) Probability $P_2 = P(a \text{ steps to a node in } I(a) \backslash I(b)) = \frac{|I(a) \backslash I(b)|}{|I(a) \cup I(b)|}$
(3) Probability $P_3 = P(b \text{ steps to a node in } I(b) \backslash I(a)) = \frac{|I(b) \backslash I(a)|}{|I(a) \cup I(b)|}$

Note that in Case 1, which we would consider the direct similarity of a and b, is described by the Jaccard coefficient. As required, the sum of these probabilities equals 1. We can then compute a similarity measure that takes the general form

$$sim_{ps}(a, b) = \sum_{i=1}^{3} P_i \cdot sim(\text{neighbors in Case } i).$$

Noting that there are $|I(a) \backslash I(b)| \cdot |I(b)|$ neighbor-pairs in Case 2 and $|I(b) \backslash I(a)| \cdot |I(a)|$ in Case 3, this produces the logical but somewhat unwieldy formula:

$$\begin{aligned} sim_{ps}(a, b) \;=\; c \Bigg[& P_1 \cdot 1 + \frac{P_2}{|I(a) \backslash I(b)|\,|I(b)|} \sum_{\substack{x \in I(a) \backslash I(b) \\ y \in I(b)}} sim_{ps}(x, y) \\ & + \frac{P_3}{|I(b) \backslash I(a)|\,|I(a)|} \sum_{\substack{x' \in I(b) \backslash I(a) \\ y' \in I(a)}} sim_{ps}(x', y') \Bigg]. \end{aligned} \tag{10}$$

MatchSim. The authors of MatchSim [Lin et al. 2009] take this amendment of random walking to its limit. They observe that when a human compares the features of two objects, a human does not select random features to see if they match. Rather, people look to see if there exists an alignment of features that produces a perfect or near-perfect matching. Therefore, their similarity measure discards the idea of random walk and replaces it with "the average similarity of the maximal matching between their neighbors" [Lin et al. 2009]:

$$sim_{ms}(a, b) = \frac{\sum_{(x,y) \in m^{\star}_{ab}} sim_{ms}(x, y)}{max(|I(a)|, |I(b)|)}, \tag{11}$$

where $m^{\star}$ represents the maximal matching. MatchSim omits the usual decay factor c, but this seems to be an idealization rather than a necessary alteration. Note that the size of the maximal matching is $min(|I(a)|, |I(b)|)$. Without loss of generality, assume that a has fewer neighbors than b. The upper bound for $sim_{ms}(a, b)$ occurs when every neighbor of a is also a neighbor of b. In this special case, $max(sim_{ms}(a, b)) = max(\frac{min(|I(a)|, |I(b)|)}{max(|I(a)|, |I(b)|)}) = \frac{|I(a) \cap I(b)|}{|I(a) \cup I(b)|}$, which is the Jaccard coefficient.

PageSim. All of the previous works are modifications of the original SimRank measure and principles. We now consider two measures that are markedly different from SimRank. We first consider PageSim [Lin et al. 2006], which not only borrows the entire PageRank computation as a starting point but also borrows the meaning of PageRank's iterative computation to devise a related computation. The canonical interpretation of PageRank is that for each step, each page sends out an equal fraction of its own importance to each of its neighbors. Its importance for the next step is the sum of the fractional importance that it received from its in-neighbors. PageSim also uses this spreading or propagating mechanism; however, rather than there being a universal importance feature that can be summed, each node begins with a distinct self-feature, which is orthogonal to every other node feature. The authors describe the propagation process as occurring over distinct paths, and they sum the contributions of each path to compute the total distribution. As long as we permit self-intersecting paths, this is equivalent to measuring the random walk destination distribution for each node after k steps. PageSim follows a multistep procedure:

(1) For each node a, define feature vector $FV(a)$. $FV_b(a)$ is the b^{th} element of $FV(a)$.
(2) Initialize all vectors: $FV_a^0(a) = PageRank(a)$. $FV_b^0(a) = 0, b \neq a$.
(3) For $t = 1$ to k iterations, $FV^t = c \cdot \sum_{a \in V} \frac{FV^{t-1}(a)}{|O(a)|}$
(4) Measure the similarity between pairs of feature vectors. In their original paper [Lin et al. 2006], the similarity measure is defined as such:

$$sim_{pg1}(a, b) = \sum_{i=1}^{n} \frac{min(FV_i(a), FV_i(b))^2}{max(FV_i(a), FV_i(b))}. \tag{12}$$

In an expanded work [Lin et al. 2007], they modify the formula to more closely resemble the Jaccard coefficient:

$$sim_{pg2}(a, b) = \frac{\sum_{i=1}^{n} min(FV_i(a), FV_i(b))}{\sum_{i=1}^{n} max(FV_i(a), FV_i(b))}. \tag{13}$$

Leicht's Vertex Similarity. The last measure that we consider addresses the other major weakness of SimRank: it considers only equal-length paths of similarity. As stated earlier, a SimRank value equals the probability that a given pair of nodes will meet *if they take steps simultaneously with the other.* That is, it would not count a case where Walker a takes three steps to reach c, and Walker b takes four steps to reach c. To address this limitation, [Leicht et al. 2005] formulate their measure from the following maxim: vertex a is similar to b if a has any neighbor c that is itself similar to b. On one hand, this statement explicitly supports asymmetrical pairs of paths. On the other hand, it assumes that being neighbors implies similarity. In Leicht's model, it follows that neighbors are somewhat similar, which describes clustering rather than role classification.

The authors did not give a catchy or convenient name to their measure, so for convenience we will call it VertexSim (notated sim_v or S_v). The initial version of VertexSim, written in matrix form, is

$$\mathbf{S_v} = \phi \mathbf{A} \mathbf{S_v} + \mathbf{I}, \tag{14}$$

where A is the adjacency matrix, and ϕ is a parameter to be determined. Solving for $\mathbf{S_v}$ and performing a power series expansion, we get

$$\mathbf{S_v} = \mathbf{I} + \phi \mathbf{A} + \phi^2 \mathbf{A}^2 + \cdots.$$

After normalizing for the expected number of paths from a to b and some simplifying approximations, the authors finally derive the following:

$$\mathbf{S_v} = \mathbf{D}^{-1} \left(\mathbf{I} - \frac{\mathbf{c}}{\lambda_1} \mathbf{A} \right)^{-1} \mathbf{D}^{-1}, \tag{15}$$

where λ_1 is the largest eigenvalue of A, and D is the degree matrix (d_{ii} = degree of node i; all other $d_{ij} = 0$). Here we have a closed form solution, which seems convenient, but we also need to invert two matrices. An iterative computation process being simpler,

Table II. Structural Similarity Measures

Measure	Formula
bibliographic coupling	$S_{bc}(a,b) = \frac{\lvert O(a) \cap O(b)\rvert}{\lvert O(a) \cup O(b)\rvert}$
co-citation	$S_{cc}(a,b) = \frac{\lvert I(a) \cap I(b)\rvert}{\lvert I(a) \cup I(b)\rvert}$
cosine	$S_{cos}(a,b) = \frac{A \cdot B}{\lvert\lvert A\rvert\rvert \; \lvert\lvert B\rvert\rvert}$
Tanimoto	$S_{tani}(a,b) = \frac{A \cdot B}{\lvert\lvert A\rvert\rvert^2 + \lvert\lvert B\rvert\rvert^2 - A \cdot B}$
weighted cosine	$S_{wcos}(a,b) = \frac{\sum_{t \in T} A(t)B(t)idf(t)}{\lvert\lvert A\rvert\rvert \; \lvert\lvert B\rvert\rvert}$
SimRank	$sim_{sr}(a,b) = \frac{c}{\lvert I(a)\rvert \lvert I(b)\rvert} \sum_{x \in I(a)} \sum_{y \in I(b)} sim_{sr}(x,y)$
SimRank++	$sim_{spp}(a,b) = c \left(\sum_{i=1}^{\lvert N(a) \cap N(b)\rvert} \frac{1}{2^i} \right) \sum_{x=1}^{N(a)} \sum_{y=1}^{N(b)} w_{ab} w_{by} sim_{spp}(x,y)$
PSimRank	$sim_{ps}(a,b) = c \left(\frac{\lvert I(a) \cap I(b)\rvert}{\lvert I(a) \cup I(b)\rvert} + \frac{\sum_{\substack{x \in I(a) \setminus I(b), \\ y \in I(b)}} sim_{ps}(x,y)}{\lvert I(a) \cup I(b)\rvert \; \lvert I(b)\rvert} + \frac{\sum_{\substack{x' \in I(b) \setminus I(a), \\ y' \in I(a)}} sim_{ps}(x',y')}{\lvert I(b) \cup I(a)\rvert \; \lvert I(a)\rvert} \right)$
MatchSim	$sim_{ms}(a,b) = \frac{\sum_{(x,y) \in m^{\star}_{ab}} sim_{ms}(x,y)}{max(\lvert I(a)\rvert, \lvert I(b)\rvert)}$
PageSim	$sim_{pg2}(a,b) = \frac{\sum_{i=1}^{n} min(FV_i(a), FV_i(b))}{\sum_{i=1}^{n} max(FV_i(a), FV_i(b))}$
VertexSim	$DS_v D = \frac{c}{\lambda_1} A(DS_v D) + I$

the authors rewrite the equation this way:

$$\mathbf{DS_v D} = \frac{\mathbf{c}}{\boldsymbol{\lambda_1}} \mathbf{A}(\mathbf{DS_v D}) + \mathbf{I}, \tag{16}$$

which we see resembles Equation (14). The authors claim that $\mathbf{DS_v D}$ can be initialized to any values such as $\mathbf{0}$ and will converge after 100 iterations or fewer.

We summarize these structural similarity measures in Table II.

3. AXIOMATIC ROLE SIMILARITY

An equivalence relation is like a simple true-false indicator: it tells us nothing about degree of similarity. The real-world need is for a measure that not only recognizes automorphic equivalence, such as Smith child/spouse/parent to Jones child/spouse/parent (Figure 1), but also tells us that a Lee child is strongly similar to a Smith or Jones child, although not as similar to a Smith or Jones parent.

To deal with this shortcoming and to clarify the problem, we first identify a list of axiomatic properties that all role similarity measures should obey.

Definition 2 (*Axiomatic Role Similarity Properties*). Given a graph $G = (V, E)$, any $sim(a, b)$ that measures the neighbor-based role similarity between vertices a and b in V should satisfy properties P1 to P5:

—P1) Range: $0 \le sim(a, b) \le 1$, for all a and b.
—P2) Symmetry: $sim(a, b) = sim(b, a)$.
—P3) Automorphism confirmation: If $a \equiv b$, $sim(a, b) = 1$.
—P4) Transitive similarity: If $a \equiv b$, then $sim(a, c) = sim(b, c)$.
—P5) Triangle inequality: $d(a, c) \le d(a, b) + d(b, c)$, where distance $d(a, c)$ is defined as $1 - sim(a, c)$.

Any node similarity measure satisfying the first four conditions (without triangle inequality) is called an **admissible role similarity measure**. Any node similarity measure satisfying all five conditions is an **admissible role similarity *metric***.

Some of these properties are representative of any node similarity measure, but Property 3 is an essential criterion that distinguishes a role similarity measure from other measures. Property 1 describes the standard normalization where 1 means fully similar and 0 means completely dissimilar (i.e., the two neighborhoods have nothing in common). That is, we should always be able to recognize a purportedly equivalent node-pair by their similarity score of 1. Property 2 indicates that similarity, like distance, must be symmetric. Property 3 expresses the requirement that if any nodes are automorphically equivalent, they must have full similarity to one another. As we discussed earlier, other definitions of role similarity are possible based on exact coloration or regular equivalence. Property 4 claims that the similarity between c and a is equal to the similarity between c and any node equivalent to a. In other words, we can define the similarity between equivalence classes or orbits: $sim(\Delta(u), \Delta(v)) = sim(u, v)$. This guarantees consistency of values at an orbit level. If Property 5 holds, the measure is metric like (i.e., it satisfies the triangle inequality).

The triangle inequality requirement is much stronger than transitivity, enforcing an *ordering* of values. In a pure metric space, the distance between any two distinct items cannot be zero. Since our automorphic equivalence property allows equivalent items to have zero distance between them, our axioms define a *pseudometric space*.

Interestingly, Property 5 (triangle inequality) implies Property 4 (transitive similarity), which we prove next. Since not all similarity measures satisfy the triangle inequality, we specify Property 4 separately.

COROLLARY 1. *If Axioms P1, P2, P3, and P5 hold, then P4 holds.*

PROOF. Let a and b be equivalent nodes. From the triangle inequality, we have $d(a, c) \le d(a, b) + d(b, c) \le d(b, c)$ because $d(a, b) = 0$. Likewise, $d(b, c) \le d(b, a) + d(a, c) \le d(a, c)$. The only way to reconcile $d(a, c) \le d(b, c)$ and $d(b, c) \le d(a, c)$ is if $d(a, c) = d(b, c)$. □

The following corollary emphasizes a fundamental difference between role similarity measures and proximity-based similarity measures: role similarity does not decrease merely because distance increases.

COROLLARY 2 (DISTANCE INDEPENDENCE). *For every finite k, there exists a graph and a pair of nodes a and b such that the distance between a and b is at least k, and* $sim(a, b) = 1$.

PROOF. Let $G(V, E)$ be a linear path graph with n edges and $n + 1$ vertices, where $n \ge k$. We label the vertices so that $V = \{v_0, v_1, \ldots, v_n\}$, and the edge set $E = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n)\}$. It is clear that nodes that are the same distance from the endpoints are automorphically equivalent and thus role equivalent. That is, $v_j \equiv v_{n-j}$, for all $0 \le j \le \lfloor n/2 \rfloor$. □

We make a final observation. Our axiomatic role similarity model can just as easily find similarities between two or more graphs, because automorphism can be extended beyond single graphs. Graph isomorphism (an equivalence relation between two graphs) is a special case of graph automorphism (an equivalence relation within a single graph) in which we declare that the two graphs are indeed a single graph that is not connected. This distance independence is absent from proximity-based node similarity measures.

3.1. Binary-Valued and Real-Valued Role Similarity Measures

Every equivalence relation σ has a corresponding binary-valued indicator function: $I_\sigma(u, v) = 1$ iff $u \equiv v$. Otherwise, $I_\sigma(u, v) = 0$. This indicator function is a admissible axiomatic role similarity metric.

THEOREM 1 (BINARY ADMISSIBILITY). *Given any equivalence relation that also satisfies automorphism confirmation (P3), its binary indicator function is an admissible similarity metric.*

PROOF. Binary values satisfy the range requirement (P1). Any equivalence relation satisfies symmetry (P2) and transitivity (P4), by definition. We prove that this indicator function satisfies the triangle inequality (P5), namely, $d(a, c) \leq d(a, b) + d(b, c)$, where $d(a, b) = 1 - sim(a, b)$, by considering all possible class assignments for a, b, and c:

Case	Description	Distances	Tri. Inequality
1	All in the same class	$d(a, c) = d(a, b) = d(b, c) = 0$	$0 \leq 0 + 0$
2	All in different classes	$d(a, c) = d(a, b) = d(b, c) = 1$	$1 \leq 1 + 1$
3	a and c in the same class	$d(a, c) = 0$	$0 \leq 1 + 1$
4	b and one other in the same class	$d(a, b) = 0$ or $d(b, c) = 0$	$1 \leq 0 + 1$

We have shown that a binary indicator function for an equivalence relation satisfies properties P1, P2, P4, and P5. Thus, if we are given that P3 is also met, then all properties are met. □

Note that automorphic equivalence, regular equivalence, and exact coloration all satisfy P3, so they are admissible metrics. Although these binary-valued similarity measures are admissible, they do not help us to understand the degree of similarity or dissimilarity. We would like a real-valued measure that ranks the degree of role similarity.

However, from the earlier discussion, we can see that the basic five axioms (P1-5) do not provide enough constraint for selecting real-valued role similarity. To achieve this, we introduce a basic property that can serve as a basic guideline for helping to measure the degree of the role similarity. Note that this property is not an axiom—that is, a real-valued role similarity may not meet its criterion. But it is intuitively appealing and can be a desirable supplement to help derive the real-valued role similarity.

Similarity Ordering: For any two vertices a and b, let d_a and d_b be their degrees, respectively. Without loss of generality, let $d_a = \min\{d_a, d_b\}$ and $d_b = \max\{d_a, d_b\}$. Consider a hypothetical pair of vertices $\overline{a}$ and $\overline{b}$ where $d_{\overline{a}} = d_a$ and $d_{\overline{b}} = d_b$. Moreover, every neighbor of $\overline{a}$ is also a neighbor of $\overline{b}$: $N(\overline{a}) \subseteq N(\overline{b})$. Given this, the role similarity of a and b should be no higher than the role similarity of $\overline{a}$ and $\overline{b}$—that is, $sim(a, b) \preceq sim(\overline{a}, \overline{b})$. In other words, $\overline{a}$ and $\overline{b}$ serve as an upper-bound benchmark to rank the role similarity. This is based on the insight that for any d_a and d_b, $\overline{a}$ is the best possible replacement for $\overline{b}$ in terms of role (assumingthat each neighbor has equal importance).

The similarity ordering criterion can be considered a supplement to automorphism confirmation (P3). It aims to deal with P3's limitation of only applying when two

Table III. Properties of Similarity Measures

Similarity Measure	Automorphism Confirmation	Transitivity	Similarity Ordering
bibliographic coupling	No	Yes	Yes
co-citation	No	Yes	Yes
SimRank	No	–	–
SimRank++	No	–	–
PSimRank	No	–	–
MatchSim	No	Yes	Yes
PageSim	Yes	Yes	No
VertexSim	No	–	–

nodes are automorphic; P3 becomes silent when nodes are not equivalent. The criterion provides guidance for the many node-pairs that are not equivalent: in what way are they not equivalent, and how can we measure this easily? There are two ways to be different: having a different number of neighbors, or having neighbors that are not quite equivalent to one another. The similarity ordering bounds the similarity based on only the former—that is, the difference between the number of neighbors. Note that here the underlying assumption is that each neighbor is considered to be an equal factor in measuring the role similarity. As our experimental results will show (Section 6), a real-valued role similarity measure satisfying this criterion (Section 4) produces superior role measure compared with the existing state-of-the-art approaches. (The alternative assumption that permits some neighbors to be more important can be a valid choice and may violate this criterion. However, it is beyond the scope of this article, and we leave it for future work.)

In the next section, we will leverage this criterion to develop a real-valued role similarity measure. Now, before presenting our proposed metric, we first examine some similarity measures proposed in earlier works.

3.2. Similarity Measures That Are Not Axiomatically Admissible

Table III categorizes the previous section's similarity measures with respect to key axiomatic role similarity properties: automorphism confirmation (P3) and transitivity (P4). We also include the similarity ordering property. We omit range (P1) and symmetry (P2) axioms because they are not cause for rejecting any of these measures.

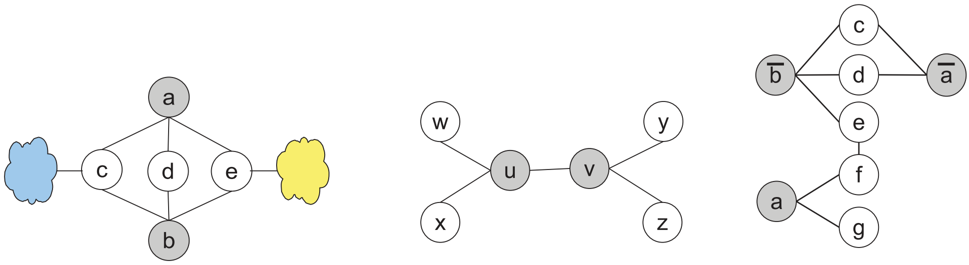
Bibliographic coupling and co-citation fail to meet the automorphism confirmation property because they only count similarity if nodes share the same neighbors. They require a path of length of 2 between the two nodes, so these measures do not satisfy the distance independence corollary.

Although SimRank and its variants seem to capture the intuition of recursive structural similarity, the random walk matching does not satisfy the basic graph automorphism condition. The SimRank similarity [Jeh and Widom 2002] between nodes u and v is based on the *average* similarity between u's neighbors and v's neighbors:

$$sim_{sr}(u,v) = \frac{(1-\beta)}{|N(u)||N(v)|} \sum_{x \in N(u)} \sum_{y \in N(v)} sim_{sr}(x,y), \quad \text{for } u \neq v,$$

$$sim_{sr}(v,v) = 1,$$

where β is a decay factor, $0 < \beta < 1$, so that the influence of neighbors decreases with distance. The original SimRank measure is for directed graphs, but it works equally well for undirected graphs. SimRank values can be computed iteratively, with successively iterations approaching a unique solution, much as PageRank [Page et al. 1999] does.



(a) Structural equivalence (b) Odd distance (c) Similarity ordering

Fig. 3. Problematic configurations for SimRank, MatchSim, and PageSim.

SimRank is not an admissible role similarity measure. We give examples where property 3 (automorphism confirmation) does not hold. In Figure 3(a), a and b have the same neighbors. By even the strictest definition (structural equivalence), a and b have the same role. However, since SimRank's initial assumption is that there is no similarity among c, d, and e, when it computes the average similarity of a and b's neighbors, it does not discover that a and b have equivalent neighborhoods. Assuming the best case where c, d, and e are in fact equivalent and using $\beta = 0.15$, $sim_{sr}(a, b)$ converges to only 0.667. Even if the neighbors are not equivalent to one another, a to b should still be equivalent, but SimRank will never discover this. We note that variants of SimRank [Antonellis et al. 2008; Fogaras and Rácz 2005; Li et al. 2009; Xi et al. 2005; Yin et al. 2006; Zhao et al. 2009] also do not meet the automorphic equivalence property for similar reasons.

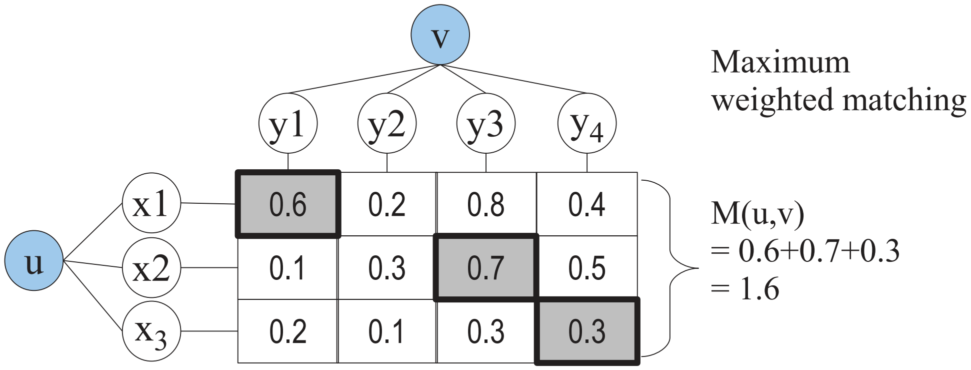
MatchSim is not an admissible role similarity measure. MatchSim seems to solve SimRank's problem of using average neighbor similarity by using maximal matching instead. If two nodes have equivalent neighborhoods, then they will have a similarity score of 1. MatchSim still falls short, however, because its initial state is too pessimistic, and its iterations do not explore all possible equivalences. Initially, each node is equivalent to itself ($sim^0(v, v) = 1$), but all distinct node-pairs are not ($sim^0(u, v) = 0$ if $u \neq v$). With each iteration, it checks for similar neighbors. In the first iteration, it detects structural equivalence, identical to co-citation and bibliographic coupling. In the second iteration, it builds on the previous similarity scores and compares neighborhoods again. However, consider Figure 3(b), in which there is an odd distance between two nodes. Nodes u and v are automorphically equivalent, but because there are no nodes that are an equal distance from both u and v, $sim_{ms}(u, v) = 0$!

PageRank-based measures, such as PageSim, meet the automorphism and transitivity axioms, so they are axiomatically admissible. However, they do not satisfy the similarity ordering property. Sharing neighbors does not guarantee that similarity will be at least as good as not sharing neighbors. Consider Figure 3(c), where every neighbor of $\overline{a}$ is also a neighbor of $\overline{b}$. However, their PageRank scores are quite different: $PR(\overline{a}) = 0.119$, but $PR(\overline{b}) = 0.175$, using $\beta = 0.9$. Vertex a has the same degree as $\overline{a}$ and has no neighbors in common with $\overline{b}$. Its PageRank score is 0.139, so $sim(a, \overline{b}) > sim(\overline{a}, \overline{b})$. We also note that in experimental evaluation (Section 6.6), PageSim performs poorly on measuring the real-world role similarity.

To our best knowledge, there is no available real-valued structural similarity measure satisfying the automorphic equivalence requirement and similarity ordering.

4. RoleSim: A REAL-VALUED ADMISSIBLE ROLE SIMILARITY

To produce an admissible real-valued role similarity measure, we face two key challenges. First, it is computationally difficult to verify the automorphic equivalence



	y1	y2	y3	y4
x1	0.6	0.2	0.8	0.4
x2	0.1	0.3	0.7	0.5
x3	0.2	0.1	0.3	0.3

Fig. 4. RoleSim(u,v) Based on Similarity of Their Neighbors.

property. Although not proven to be NP complete, the graph automorphism problem has no known polynomial algorithm [Fortin 1996]. Second, all of the existing real-valued role similarity measures have problems dealing with even simple conditions such as structural equivalence (Section 3.2). To meet these challenges, we take the following approach: given an initial simplistic but admissible role similarity measurement for each pair of nodes, refine the measurement by expressing similarity in terms of neighboring values while maintaining the automorphic and structural equivalence properties. Using this approach, we formally introduce RoleSim, the first admissible real-valued role similarity measure (metric) and its associated properties. It also satisfies the similarity ordering property.

4.1. RoleSim Definition

Given a graph $G = (V, E)$, the RoleSim measure realizes the recursive node structural similarity principle “two nodes are similar if they relate to similar objects” as follows.

Definition 3 (*RoleSim Metric*). Given two vertices u and v, where $N(u)$ and $N(v)$ denote their respective neighborhoods and d_u and d_v denote their respective degrees, then

$$RoleSim(u, v) = (1 - \beta) \max_{M(u,v)} \frac{\sum_{(x,y)\in M(u,v)} RoleSim(x, y)}{d_u + d_v - |M(u, v)|} + \beta. \qquad (17)$$

where $x \in N(u)$, $y \in N(v)$, and $M(u, v)$ is a **matching** between $N(u)$ and $N(v)$—that is, $M(u, v) = \{(x, y)|x \in N(u), y \in N(v),$ and no other $(x', y') \in M(u, v),$ s.t., $x = x'$ or $y = y'\}$. The parameter β is a decay factor, $0 < \beta < 1$.

The decay factor, similar to the one used in PageRank [Page et al. 1999], both dampens the recursive effect and guarantees a minimal RoleSim score of β. We will sometimes abbreviate *RoleSim*(u, v) as $R(u, v)$. **R** refers to the entire matrix of values. Figure 4 illustrates the matching process. Vertex u has three neighbors $(x_1, x_2, x3)$, and v has four neighbors (y_1, y_2, y_3, y_4). The (x, y) grid is the subset of the RoleSim matrix of values corresponding to the pairings of neighbors of these two vertices. A *matching* selects one cell per row and column. If the number of rows differs from the number of columns, then the matching size is limited to $|M(u, v)| = min(d_u, d_v)$. A *maximal matching* is a matching where the total value of selected cells is maximum. In contrast, SimRank computes the average of every cell in the neighbor grid.

4.1.1. Relation to Jaccard and Tanimoto Coefficients. RoleSim employs a generalization of the Jaccard coefficient, which measures the commonality between two sets A and B as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Previous works [Fogaras and Rácz 2005] have used this index to compare node neighborhoods; several variants exist [Melançon and Sallaberry 2008]. Our denominator is similar to that of the Tanimoto coefficient [Tanimoto 1958], which measures similarity between multisets or between vectors. In our generalization, however, sets A and B are not vectors and need not share any common elements; instead, there is a weighted matching M between *similar* elements in A and B—that is, $(a, b) \in M, a \in A, b \in B$. Let $r(a, b) \in [0, 1]$ record the similarity between a and b.

Definition 4 (*Generalized Jaccard Coefficient*). The generalized Jaccard coefficient measures the similarity between two sets A and B under matching M, defined as

$$J(A, B|M) = \frac{\sum_{(a,b)\in M} r(a, b)}{|A| + |B| - |M|}. \tag{18}$$

The original Jaccard coefficient is a special case that uses the following matching M: let $r(x, y) = 1$ if $x = y$; otherwise, 0. Then define $M = \{r(x, x)|x \in A, x \in B\}$. Thus, the generalized Jaccard coefficient $J(A, B|M)$ reduces to $J(A, B)$. Comparing Equations (17) and (18), we see that the heart of $RoleSim(u, v)$ is equivalent to the maximum of the generalized Jaccard coefficient between $N(u)$ and $N(v)$, among all matchings $M(u, v)$. Then,

$$RoleSim(u, v) = (1 - \beta) \max_{M(u,v)} J(N(u), N(v)|M(u, v)) + \beta. \tag{19}$$

4.1.2. Relation to Weighted Matching. The definition and significance of $RoleSim(u, v)$ is closely related to *maximal weighted matching*. In our case, the matching is between the neighboring nodes of u and v.

Definition 5 (*Maximal Neighborhood Matching* $\mathbb{M}(\mathbf{u}, \mathbf{v})$). Let $R(x, y)$ be a similarity score between any two nodes x and y (0 if no score exists). Given two nodes u and v, their neighborhood matching $M(u, v)$ is a weighted bipartite matching between neighbor sets $N(u)$ and $N(v)$ where the weights are the $R(x, y)$ scores. The weight of the matching is $w(M) = \sum_{(x,y)\in M} R(x, y)$. A maximal matching $\mathbb{M}(u, v)$ is an M with maximum weight.

Using this, we can represent $RoleSim(u, v)$ in terms of maximal weighted matching $\mathbb{M}$. In Figure 4, the shaded cells represent the maximal matching: $0.7 + 0.6 + 0.3 = 1.6$.

THEOREM 2 (MAXIMAL WEIGHTED MATCHING). *The RoleSim between nodes u and v corresponds linearly to the maximal weighted matching* $\mathbb{M}$ *for the bipartite graph* $(N(u), N(v), N(u) \times N(v))$, *with each edge* $(x, y) \in N(u) \times N(v)$ *having the weight* $RoleSim(x, y)$:

$$RoleSim(u, v) = (1 - \beta)\frac{w(\mathbb{M})}{\max(d_u, d_v)} + \beta. \tag{20}$$

PROOF. We need to show that Equations (17) and (20) are equivalent. Without loss of generality, let $d_u \geq d_v$. First, we show that *the cardinality of the maximal weighted matching* $|\mathbb{M}| = \min(d_u, d_v) = d_v$. It cannot be greater, because there are insufficient elements in d_v. It cannot be smaller, because if it were, there would exist an available edge between an uncovered node in d_u with one in d_v. Adding this edge would increase the matching (every edge has weight $\geq \beta$). If $|\mathbb{M}| = \min(d_u, d_v)$, it follows that $d_u + d_v - |M| = \max(d_u, d_v)$. Thus, the denominators in Equations (17) and (20) are constant and identical. It is then a trivial observation that the numerators are in fact the same. Therefore, the maximal value for the entire Equation (17) is the same as the value in Equation (20). □

Theorem 2 not only shows the key equilibrium of role similarities between pairs of nodes in a graph G, but it also shows that RoleSim may be computed using existing maximal matching algorithms. Finally, given Theorem 2, we can state the following lemma:

LEMMA 4.1. *RoleSim satisfies the similarity ordering criterion.*

PROOF. Consider any two vertices a and b where a's degree is not larger: $d_a \leq d_b$. Let $\overline{a}$ and $\overline{b}$ be hypothetical vertices where $d_{\overline{a}} = d_a$, $d_{\overline{b}} = d_b$. Moreover, every neighbor of $\overline{a}$ is a neighbor of $\overline{b}$: $N(\overline{a}) \subseteq N(\overline{b})$. Due to this maximal sharing of neighbors, $RoleSim(\overline{a}, \overline{b}) = (1-\beta)\frac{d_a}{d_b} + \beta$. In general, however, $w(\mathbb{M}) \leq d_a$. Thus, $RoleSim(a, b) \leq (1-\beta)\frac{d_a}{d_b} + \beta = RoleSim(\overline{a}, \overline{b})$. □

4.2. RoleSim Computation

RoleSim values can be computed iteratively and can be guaranteed to converge, just as in PageRank and SimRank. First, we outline the iterative procedure. In the next section, we prove that the calculated values comprise an admissible role similarity metric.

Step 1: Let the initial matrix of RoleSim scores be $\mathbf{R^0}$, estimated but admissible scores between any pair of nodes in G. Simple initialization schemes are described in Section 4.4.

Step 2: Compute the k^{th} iteration $\mathbf{R^k}$ scores from the $(k-1)^{th}$ iteration's values, $\mathbf{R^{k-1}}$. Specifically, for any nodes u and v,

$$R^k(u, v) = (1-\beta) \max_{M(u,v)} \frac{\sum_{(x,y)\in M(u,v)} R^{k-1}(x, y)}{d_u + d_v - |M(u, v)|} + \beta. \tag{21}$$

Based on Theorem 2, we compute Equation (21) by finding the maximal weighted matching in the weighted bipartite graph $(N(u), N(v), N(u) \times N(v))$ with each edge $(x, y) \in N(u) \times N(v)$ having weight $R^{k-1}(x, y)$).

Step 3: Repeat Step 2 until $|\mathbf{R^k} - \mathbf{R^{k-1}}| <$ some threshold δ for each pair of nodes in G.

THEOREM 3 (GUARANTEED TERMINATION). *For any admissible set of initial RoleSim^0 values and any termination threshold $\delta > 0$, the change in RoleSim values between iterations will become arbitrarily small—for example, for any (u, v) pair,*

$$\lim_{k\to\infty} |RoleSim^k(u, v) - RoleSim^{k-1}(u, v)| < \delta. \tag{22}$$

This can be proven by showing that the sequence of maximum absolute differences between any $\mathbf{R^k}(u, v)$ and $\mathbf{R^{k+1}}(u, v)$, for $k = 1, 2, \ldots$, is a nonnegative geometric sequence monotonically decreasing and converging to 0. The detailed proof is in Appendix A.

Unlike PageRank and SimRank, which converge to values independent of the initialization, RoleSim values are sensitive to the initialization. Rather than being a disadvantage, this sensitivity provides the necessary relaxation to compute automorphic role similarity in polynomial time by utilizing the initialization as prior knowledge.

4.3. Admissibility of RoleSim

Here we present one of the key contributions of this article: the axiomatic admissibility of RoleSim. If the initial computation is admissible, and because the iterative computation of Equation (20) maintains admissibility (i.e., is an invariant transform of the axiomatic properties), then the final measure is admissible.

THEOREM 4 (INVARIANT TRANSFORMATION). *If the k^{th} iteration $RoleSim^k$ is an admissible role similarity metric, then so is $RoleSim^{k+1}$.*

For each axiomatic property P, we must show "If the k^{th} iteration $RoleSim^k$ satisfies Axiom P, then so does $RoleSim^{k+1}$." Properties 1 (range) and 2 (symmetry) are trivially invariant, so we will focus on the other three.

Automorphism Confirmation Invariance Proof: For nodes where $u \equiv v$, there is a permutation σ of vertex set V, such that $\sigma(u) = v$, and any edge $(u, x) \in E$ iff $(v, \sigma(x)) \in E$. This indicates that σ provides a one-to-one equivalence between nodes in $N(u)$ and $N(v)$. In addition, u and v have the same number of neighbors—that is, $d_u = d_v$. So, it is clear that the maximal weighted matching $\mathbb{M}$ in the bipartite graph $(N(u), N(v), N(u) \times N(v))$ selects $d_u = d_v$ pairs of weight 1 each. Thus, $RoleSim^{k+1}(u, v) = (1-\beta)\frac{w(\mathbb{M})}{\max(d_u, d_v)} + \beta = (1-\beta)\frac{d_u \cdot 1}{d_u} + \beta = 1$. □

Transitive Similarity Invariance Proof: Assume that transitivity holds for iteration k: for any $a \equiv b$, $c \equiv d$, $RoleSim^k(a, c) = RoleSim^k(b, d)$. Denote the maximal weighted matching between $N(a)$ and $N(c)$ as $\mathbb{M}$. Since there is a one-to-one equivalence correspondence σ between neighborhoods $N(a)$ and $N(b)$ and a one-to-one equivalence correspondence σ' between $N(c)$ and $N(d)$, we can construct a matching $\mathbb{M}'$ between $N(b)$ and $N(d)$ as follows: $\mathbb{M}' = \{(\sigma(x), \sigma'(y)) | (x, y) \in \mathbb{M}\}$. Since transitive similarity holds for $RoleSim^k$, we have $RoleSim^k(x, y) = RoleSim^k(\sigma(x), \sigma'(y))$. Thus, $w(\mathbb{M}') = w(\mathbb{M})$, and

$$\begin{gathered}(1-\beta)\frac{w(\mathbb{M})}{\max(d_a, d_c)} + \beta = (1-\beta)\frac{w(\mathbb{M}')}{\max(d_b, d_d)} + \beta \\ RoleSim^{k+1}(a, c) = RoleSim^{k+1}(b, d). \quad \square\end{gathered}$$

Triangle Inequality Invariance Proof: For iteration k, for any nodes a, b, and c, $d^k(a, c) \leq d^k(a, b) + d^k(b, c)$, where $d^k(a, b) = 1 - RoleSim^k(a, b)$. We must prove that this inequality still holds for the next iteration: $d^{k+1}(a, c) \leq d^{k+1}(a, b) + d^{k+1}(b, c)$.

Observation: *If there is any matching M between $N(a)$ and $N(c)$ that satisfies* $1 - ((1-\beta)\frac{w(M)}{d_c} + \beta) \leq d^{k+1}(a, b) + d^{k+1}(b, c)$*, then* $d^{k+1}(a, c) \leq d^{k+1}(a, b) + d^{k+1}(b, c)$. This is because $\frac{w(M)}{d_c} \leq \frac{w(\mathbb{M})}{d_c}$, where $\mathbb{M}$ is the maximal weighted matching between $N(a)$ and $N(c)$, and thus, $1 - ((1-\beta)\frac{w(M)}{d_c} + \beta) \geq 1 - ((1-\beta)\frac{w(\mathbb{M})}{d_c} + \beta) = d^{k+1}(a, c)$.

We break down the proof into three cases:
Case 1. $(d_b \leq d_a \leq d_c)$,
Case 2. $(d_a \leq d_b \leq d_c)$, and
Case 3. $(d_a \leq d_c \leq d_b)$.

Case 1: Since d_b is smallest, this sets the size for the maximal neighborhood matchings: $|\mathbb{M}(a, b)| = |\mathbb{M}(b, c)| = d_b$. Define candidate matching M between $N(a)$ and $N(c)$ as $M = \{(x, z) | (x, y) \in \mathbb{M}(a, b) \wedge (y, z) \in \mathbb{M}(b, c)\}$. Then, using our earlier observation:

$$\begin{aligned} & d^{k+1}(a, b) + d^{k+1}(b, c) - \left(1 - (1-\beta)\frac{w(M)}{d_c} - \beta\right) \\ & = (1-\beta)\left[-\frac{w(\mathbb{M}(a, b))}{d_a} - \frac{w(\mathbb{M}(b, c))}{d_c} + \frac{w(M)}{d_c}\right] + 1 - \beta \\ & = (1-\beta)\left[\frac{d_b - w(\mathbb{M}(a, b))}{d_a} - \frac{d_b}{d_a} + \frac{d_b - w(\mathbb{M}(b, c))}{d_c}\right. \\ & \quad \left. - \frac{d_b}{d_c} - \frac{d_b - w(M)}{d_c} + \frac{d_b}{d_c}\right] + 1 - \beta \\ & \geq (1-\beta)\left[1 - \frac{d_b}{d_a} + \frac{\sum_{(x,y)\in\mathbb{M}(a,b)}(1 - R^k(x, y))}{d_c}\right. \end{aligned}$$

$$
\left. + \frac{\sum_{(y,z)\in\mathbb{M}(b,c)}(1 - R^k(y,z))}{d_c} - \frac{\sum_{(x,z)\in M}(1 - R^k(x,z))}{d_c} \right]
$$

$$
\geq (1-\beta)\left[\frac{\sum_{(x,y,z)}(d^k(x,y) + d^k(y,z) - d^k(x,z))}{d_c}\right] \geq 0,
$$

where (x, y, z) means $(x, y) \in \mathbb{M}(a, b)$, $(y, z) \in \mathbb{M}(b, c)$, and $(x, z) \in M$. □

Cases 2 and 3 can be proven by a similar technique; the complete proof is in Appendix A.

By combining the admissible initial configurations given in Section 4.4 with Theorem 4 on invariance, we have shown that the iterative RoleSim computation generates a real-valued, admissible role similarity measure.

THEOREM 5 (ADMISSIBILITY). *If the initial $RoleSim^0$ is an admissible role similarity measure, then at each k^{th} iteration, $RoleSim^k$ is also admissible. When RoleSim computation converges, the final measure $\lim_{k\to\infty} RoleSim^k$ is admissible.*

4.4. Initialization

According to Theorem 5, an initial admissible RoleSim measurement $\mathbf{R^0}$ is needed to generate the desired real-valued role similarity ranking. What initial admissible measures or prior knowledge should we use? We consider three schemes:

(1) **ALL-1:** $R^0(u, v) = 1$ for all u, v.
(2) **Degree-Binary (DB):** If two nodes have the same degree ($d_u = d_v$), then $R^0(u, v) = 1$; otherwise, 0.
(3) **Degree-Ratio (DR):** $R^0(u, v) = (1 - \beta)\frac{min(d_u,d_v)}{max(d_u,d_v)} + \beta$.

These schemes come from the following observation: *nodes that are automorphically equivalent have the same degree*. Equal degree is a necessary, but not sufficient, condition for automorphism. This observation is key to RoleSim: degree affects both the size of a maximal matching set and the denominator of the Jaccard coefficient.

We make the following interesting observations about these initialization schemes.

LEMMA 4.2. *Let $\mathbf{R^1}(ALL - 1)$ be the matrix of RoleSim values at the first iteration after $\mathbf{R^0} = \mathbf{1}$ (ALL-1 initialization). Let $\mathbf{R^0}(DR)$ be the matrix of RoleSim initialized by the DR scheme. Then, $\mathbf{R^1}(ALL - 1) = \mathbf{R^0}(DR)$.*

This lemma can be easily derived by following the definition of RoleSim formula. Basically, the DR is exactly equal to the RoleSim state one iteration after ALL-1 initialization. Thus, ALL-1 and DR generate the same final results. The simple formula for DR is much faster than neighbor matching, so DR is essentially one iteration faster. On the other hand, we may consider the simple ALL-1 scheme to be sufficient, since it works as well as the more sophisticated DR. After the simple ALL-1 initialization, RoleSim's maximal matching process automatically discriminates between nodes of different degree and progressively learns the differences among neighbors as it iterates.

THEOREM 6 (ADMISSIBLE INITIALIZATION). *ALL-1, DB, and DR are all admissible role similarity metrics.*

PROOF. It is easy to see that ALL-1 degenerately satisfies all of the axioms of a role similarity metric. For DR, Lemma 4.2 shows that its matrix of values is equivalent to ALL-1's matrix after one RoleSim iteration. Since RoleSim iterations will preserve the admissibility of a metric (Theorem 4), DR is also a metric. For DB, it trivially satisfies range (P1), symmetry (P2), and automorphism confirmation (P3). For transitive similarity (P4), we only need to show that $R^0(u, v)$ depends only on class membership. Class is defined by degree, and the measurement clearly depends only on degree. Finally, because DB is a binary indicator of equivalence, Theorem 1 states that it is a role similarity metric. □

Note that SimRank's and MatchSim's initialization ($sim^0(u, v) = 1$ iff $u = v$) is NOT admissible, because it sets the initial value of any potentially equivalent node-pairs to 0. SimRank and MatchSim iterations try to build up from zero. However, due to problems with structural equivalence and odd-length paths that we noted, they will never increase the value enough to discover all of the equivalent pairs that were neglected at the start.

In addition, both ALL-1 and DR initialization have the following convergence property, which is stronger than our earlier guaranteed termination property:

THEOREM 7 (MONOTONE CONVERGENCE). *If ALL-1 initialization is used, each RoleSim value is monotonically decreasing (or nonincreasing):* $\mathbf{R^{k+1}}(u, v) \leq \mathbf{R^k}(u, v)$ *for all k.*

PROOF. At any iteration, the RoleSim value for any (u, v) is the maximal matching of its neighbors. The value can increase only if some neighbor matchings increase. If no value increased in the previous iteration, then no value can increase in the current iteration. In the first iteration after ALL-1, clearly no value increases. Therefore, no value ever increases. Any function that decreases monotonically and has a lower bound will converge. The lower-bound value is β, so RoleSim with ALL-1 initialization is guaranteed to converge. □

Indeed, this monotone convergence property can be generalized into the following format: *if* $\mathbf{R^1} \leq \mathbf{R^0}$ *(that is, for every* (u, v) *pair,* $\mathbf{R^1}(u, v) \leq \mathbf{R^0}(u, v)$*), then* $\mathbf{R^{k+1}} \leq \mathbf{R^k}$. Note that the DB initialization scheme does not have this property. In our experiments, we make an empirical comparison of these initialization schemes.

4.5. Computational Complexity

Given n nodes, we have $O(n^2)$ node-pair similarity values to update for each iteration. For each node-pair, we must perform a maximal weighted matching. For weighted bipartite graph $(N(u), N(v), N(u) \times N(v))$, the fastest algorithm based on augmenting paths (Hungarian method [Kuhn 1955]) can compute the maximal weighted matching in $O(x(x \log x + y))$, where $x = |N(u)| + |N(v)|$, and $y = |N(u)| \times |N(v)|$.

A fast greedy algorithm offers a $\frac{1}{2}$-approximation of the globally optimal matching in $O(y \log y)$ time [Avis 1983]. Furthermore, if an equivalence matching exists (i.e., $w(\mathbb{M}) = \max(d_u, d_v)$), the greedy method will find it. This is important, because it means that a greedy RoleSim computation still generates an admissible measure. Using greedy neighbor matching, the time complexity of RoleSim is $O(kn^2d')$, for k iterations, where d' is the average of $y \log y$ over all vertex-pair bipartite graphs in G. The space complexity is $O(n^2)$. In the next section, we will introduce an approach for reducing both the time and memory cost.

5. ICEBERG RoleSim: A SCALABLE ALGORITHM

Node similarity ranking in general is computationally expensive because we need to compute the similarity for $\binom{n}{2} = O(n^2)$ node-pairs. A graph with 100,000 nodes needs about 40GB memory to simply maintain the similarity values, assuming 8 bytes per value. Indeed, this is a major problem for almost all node similarity ranking algorithms. However, in most applications, we are interested only in the *highest* similarity pairs, which typically compose only a very small fraction of all pairs. Thus, in order to improve the scalability of RoleSim, we ask the following question: *can we* *identify the high-similarity pairs without computing all pair similarities?* Formally, we consider the following question:

Definition 5.1 (*Iceberg RoleSim*). Given a threshold θ, the Iceberg RoleSim problem is to discover all (u, v) pairs for which $RoleSim(u, v) \geq \theta$ and then approximate their RoleSim scores.

The goal is to identify and compute those high-similarity pairs without materializing the majority of the low-similarity pairs. To solve *Iceberg RoleSim*, we consider a two-step approach: (1) use pruning rules to rule out pairs whose score must be less than θ, and (2) apply RoleSim iterative computation to the remaining candidate pairs. Since RoleSim computation must match all neighbor-pairs $(N(u) \times N(v))$ of a candidate pair (u, v), we have to handle neighbor-pairs (such as x, y) that are not themselves candidate pairs. Here, we employ upper and lower bounds for estimating RoleSim values for the noncandidate pairs.

Upper and Lower Bound for RoleSim:

LEMMA 5.2. *Given nodes u, v and without loss of generality, $d_u \geq d_v$, if $d_v \leq \theta d_u$, then similarity $R(u, v) \leq (1 - \beta)\theta + \beta$.*

PROOF. $R(u, v) = (1 - \beta)\frac{w(\mathcal{M})}{d_u} + \beta \leq (1 - \beta)\frac{d_v}{d_u} + \beta$. □

Given this, assuming $d_u \geq d_v$, since matching $0 \leq w(\mathcal{M}) \leq d_v$, then $R(u, v)$ is in the range $[\beta, (1 - \beta)\frac{d_v}{d_u} + \beta]$. Furthermore, to facilitate our discussion, we further define $\theta' = (\theta - \beta)/(1 - \beta)$. Now, we introduce the following *pruning rules* to filter out those pairs whose RoleSim cannot be greater than or equal to threshold θ, without knowing their exact *RoleSim* scores (without loss of generality, let $d_u \geq d_v$):

(1) If $d_v < \theta' d_u$, then $R(u, v) < \theta$.
(2) If maximal matching weight $w(\mathcal{M}) < \theta' d_u$, then $R(u, v) < \theta$.
(3) Assume that neighbor lists $N(u)$ and $N(v)$ are sorted by degree, with d_1^u and d_1^v being the degrees of the first items. The maximum possible similarity of this pair is $m_{11} = (1 - \beta)\frac{min(d_1^u, d_1^v)}{max(d_1^u, d_1^v)} + \beta$. If the shorter list has the smaller degree ($d_1^v \leq d_1^u$), and if $m_{11} + d_v - 1 < \theta' d_u$, then $R(u, v) < \theta$.

Rule 1 is just a restatement of Lemma 5.2. Rule 2 is based on the upper bound of RoleSim value. Rule 3 requires more explanation: continuing from Rule 2, we begin to consider all pairings of neighbors. Because d_v is the shorter list, every member must contribute to the final matching. Either m_{11} will be in the matching or not. If it is, then an upper bound for $\mathcal{M}$ is if every remaining pair has weight 1, yielding $m_{11} + (d_v - 1)$. Additionally, because the lists are sorted, $d_1^v/d_1^u \geq d_1^v/d_x^u$, for $x > 1$. So, if m_{11} is too small to satisfy Rule 2, then all pairings using d_1^v are too small. This rule allows us to short circuit the full neighbor matching.

5.1. Iceberg Algorithm

We now outline our approach, which is formalized in Algorithm 1. To generate the initial iceberg hash map, we sort nodes by degree (line 3) and sort each node's list of neighbors by degree (lines 4 to 6). The first sort allows us to consider only those node-pairs that are sufficiently similar in degree (line 8, pruning Rule 1). We compute the estimated similarity for the first pair of neighbors. Note that this estimation formula is the same as DR initialization. If this weight is below the limit defined in Rule 3, we terminate this pair's candidacy and move on (lines 9 through 12). Otherwise, compute the remainder of neighbor-pair initial similarities, and perform a maximal matching. If the matching weight exceeds the θ' minimum bound (Rule 2), then this node-pair

ALGORITHM 1: IcebergRoleSim($G(V, E)$, θ, β, α)

```
1:  $H \leftarrow$ empty hash table indexed by node-pair ID $(u, v)$;
2:  $d(v) \leftarrow$ degree of $v$;
3:  Sort vertices $V$ by degree;
4:  for all $v \in V$ do
5:      $D^v = \{d_1^v, d_2^v, \ldots, d_{d(v)}^v\} \leftarrow$ degrees of neighbors of $v$, sorted by increasing order;
6:  end for
7:  for all $u \in V$ do
8:      for all $v \in V, \theta' d(u) \leq d(v) \leq d(u)$ (Rule 1) do
9:          $m_{11} \leftarrow (1-\beta)\frac{min(d_1^u, d_1^v)}{max(d_1^u, d_1^v)} + \beta$;
10:         if $d_1^v \leq d_1^u$ and $d_v - 1 + M_{11} < \theta' d_u$ then
11:             Skip to the next $v$; (Rule 3)
12:         end if
13:         Compute maximal matching weight $w(\mathcal{M})$;
14:         if $w(\mathcal{M}) \geq \theta' d(u)$ (Rule 2) then
15:             Insert $H(u, v) \leftarrow (1-\beta) w(\mathcal{M})/d(u) + \beta$;
16:         end if
17:     end for
18: end for
19: Perform iterative RoleSim on $H$. For neighbor-pairs $\notin H$, use $\tilde{R}(x, y) = \alpha(1-\beta)N_x/N_y + \beta$.
```

and its similarity are inserted into the hash table (lines 13 through 16). After iterating through all qualified node-pairs, we have our full hash table. We now perform RoleSim iterations, but only on members of the table, which is orders of magnitude smaller than a complete similarity matrix. When a noncandidate pair's value is needed (as a neighbor-pair of a candidate pair), we apply the following estimate based on its lower and upper bound (assuming $d_u \geq d_v$):

$$\tilde{R}(u, v) = \alpha(1-\beta)\frac{d_v}{d_u} + \beta, \text{ where } 0 \leq \alpha \leq 1.$$

In the experimental evaluation, we will empirical study the effect of α on the estimation accuracy.

5.2. Iceberg Computational Complexity

Iceberg RoleSim's time and memory requirements are best understood as multiplicative improvement factors over standard RoleSim. For memory complexity, the factor is mainly the reduction in the number of stored node-pairs due to pruning. Let us define pruning factor p as the ratio between the number of node-pairs stored in standard RoleSim and the number of top similarity values stored in an Iceberg hash table. Thus, $p = \binom{n}{2}/|H| \approx \frac{n^2}{2|H|}$. Larger values mean a lower space requirement. Each hash table entry is a little more expensive than an equivalent similarity matrix entry, because it must store the vertex IDs u and v as well as $R(u, v)$. Consequently, the memory improvement factor is somewhat less than p.

The hash table size $|H|$ varies according to the graph's structural characteristics and the pruning threshold θ. In our experiments with random power law graphs, we measured values of p ranging from 33 to 814. The largest reductions, by nearly three orders of magnitude, occur when higher values of θ are applied to graphs that are denser and larger, exactly when the improvement is needed most. This may be because these graphs have more internal structural diversity and thus less role similarity.

There are two major contributing factors to time complexity: the initial pruning and estimation of values, followed by the iterative refinement of values. The time complexity of the second stage (line 19) is p times faster than the time complexity of standard RoleSim, because Iceberg RoleSim iterates over the pruned hash tables instead of a complete $n \times n$ matrix. Moreover, the algorithm typically converges after fewer iterations, because there are fewer values being updated. Let us take a closer look at the initial stage, as detailed in Algorithm 1. Sorting the vertices by degree (Step 3) is a bucket sort, so it requires only $O(n)$ time. Sorting the neighbors of each node by degree (Steps 4 through 6) is $O(n \cdot \hat{d} \log \hat{d})$, where $\hat{d}$ = expected (weighted average) node degree. Next comes the nested for loops (lines 7 through 18), which implement a modified RoleSim iteration, incorporating pruning and estimation of values. Because the pruning is occurring amidst this loop, the time complexity falls in between that of a full standard RoleSim iteration and a p-pruned iteration. In practice, it is closer to a fully p-pruned iteration. So, the overall time complexity is $O(n \cdot \hat{d} \log \hat{d} + k'|H|d')$, where d' is the same as in Section 4.5.

6. EXPERIMENTAL EVALUATION

In this section, we experimentally investigate the ranking ability and performance of the RoleSim algorithm for computing role similarity metric values. We compare RoleSim to several state-of-the-art node similarity algorithms, analyze the effect of different initialization schemes, and measure the scalability of Iceberg RoleSim. Specifically, we focus on the following questions:

(1) How do different initialization schemes perform in terms of their final RoleSim score and computational efficiency?
(2) Do node-pairs with high RoleSim scores actually have similar network roles? For any two nodes known to have similar network roles, do they receive high role similarity scores?
(3) How much less memory and time does Iceberg RoleSim use, and how closely does its rankings match that of standard RoleSim?

Clearly, the ideal validation study requires an explicit role model and role similarity measure, which often do not exist. In the following study, we utilize a well-known role-related random graph model and external measures of real datasets that provide strong role indication for these evaluations.

We set $\beta = 0.1$ for both RoleSim and SimRank, defining convergence to be when values change by less than 1% of their previous values. We ran several RoleSim tests with both exact matching and greedy matching. The results were nearly identical (>90% of cells have no difference; maximum difference was small), so we focus on greedy matching from here on. We implemented the algorithms in C++. The large graph scalability tests were performed on Linux servers with 3.2GHz quad-core Xeon CPUs, 2MB cache, and 16MB RAM.

For our tests, we use three types of graphs:

- **BL:** The probabilistic blockmodel [Wang and Wong 1987], where each block is generally considered to be corresponding to a role [White et al. 1976]. Here, nodes are partitioned into blocks. Each node in block i has probability p_{ij} of linking to each node in block j. Thus, the underlying blockmodel may serve as the ground truth for testing role similarity.
- **SF:** Large Scale-Free random graphs[2] are used for testing scalability of the Iceberg RoleSim computation.
- Real-world networks, with a measurable feature similar to social role, are used for validating RoleSim performance.

[2] http://pywebgraph.sourceforge.net/.

Table IV. Performance Comparison of Initialization Methods

Performance Relative to	DR	DB Init.		
ALL-1 Initialization	Init.	Min.	Avg.	Max.
Difference in percentile rank	none	0.14%	0.38%	11.17%
Pearson correlation coefficient	1	0.9994	0.9998	0.9999
Relative execution time	≈0.9	0.32	0.52	0.80
Relative # iterations	1 fewer	0.38	0.58	0.88

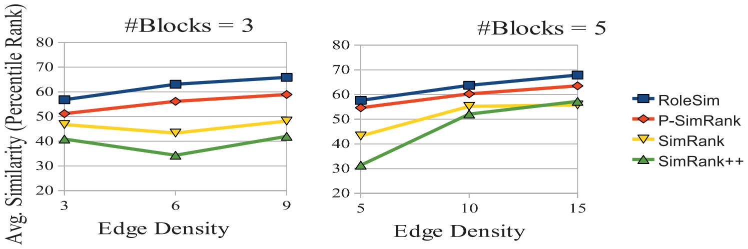


Fig. 5. Average similarity ranking for nodes in the same block.

6.1. Comparing Initialization

In Section 4.4, we discussed that DR initialization generates the same results as ALL-1 by short cutting the first iteration. This reduces the computation time by roughly 10%. Now we ask: does DB initialization (binary indicator that equals 1 when degrees $d_u = d_v$) give similar results, quickly?

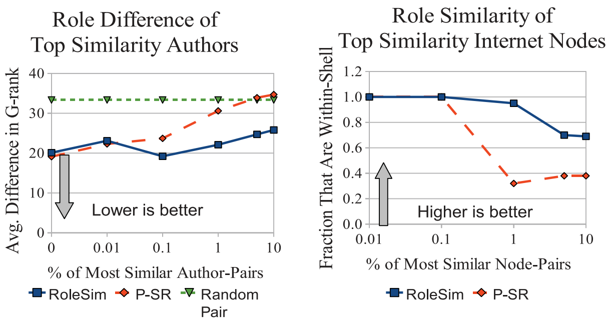
We ran RoleSim using both ALL-1 and DB on 12 graphs, some scale free and some blockmodel, having 500 to 10,000 nodes, and edge densities from 1 to 10. We then converted values to percentile ranking, where 100% means the highest value, and 50% is the median value. Test results are summarized in Table IV. The high correlation coefficient means the rankings are virtually identical, so the rankings are not very sensitive to the initialization method. Moreover, DB took 20% from 68% less time to converge. Overall, *DB* seems to be the preferred initialization scheme in terms of computational efficiency. Thus, we adopt it for the rest of the experiments.

6.2. General Role Detection

How well does RoleSim discover roles in complex graphs? Specifically, given a ground-truth knowledge of roles, do nodes having similar roles have high scores? To answer this question, we generated probabilistic blockmodel graphs, where blocks behave like "noisy" roles, due to sampling variance. We generated graphs with $N = 1{,}000$ nodes and either three or five blocks. We varied the edge density $\frac{|E|}{|V|}$, with higher densities for graphs with more blocks. The size of each block and the p_{ij} values were randomized; we generated three random instances for each graph class. We compared RoleSim to the state-of-the-art SimRank, SimRank++ [Antonellis et al. 2008], and P-SimRank [Fogaras and Rácz 2005].

For each measure and trial, we ranked the similarity scores. This serves to normalize the scoring among the four measures. Then, for each graph, we computed the average ranking of all pairs of nodes within the same block. We then averaged the three trials for each graph class.

Our results (Figure 5) show that RoleSim outperforms all other algorithms across all the tested conditions. None of the algorithms score perfectly, due to the inherent edge distribution variance of the probabilistic model. P-SimRank is better than SimRank, perhaps because it uses Jaccard coefficient weighting, a step toward our



(a) Top coauthors (b) Top Internet nodes

Fig. 6. Similarity of nodes for top-ranked node-pairs.

RoleSim approach. Accuracy takes time. SimRank and SimRank++ run at the same speed. P-SimRank is about 1.5 to 2 times slower, and standard RoleSim is about twice as slow as SimRank.

6.3. Real Dataset: Coauthor Network

Historically, structural role has not been measured precisely in large graphs, so it is difficult to find datasets with established ground-truth roles. In this experiment, we use a coauthor network and take author impact, as measured by G-index and H-index scores, as the ground truth. Based on recent studies, we expect network role in a coauthor network to correlate well to author impact and thus serve as a predictive measure. Earlier investigations [Newman 2004; Otte and Rousseau 2002] observed structural patterns of collaboration in coauthor networks and speculated that hubs (nodes with high degree) and connectors (nodes with high betweenness) would be likely candidates for high-impact authors. Recently, Yan and Ding [2009] discovered significant correlation between the number of citations and certain coauthor network measures: betweenness centrality and PageRank similarity.

Our first dataset [Tang et al. 2008] is a coauthor network of 2,000 database researchers. Two authors are linked if they coauthored a paper from 2003 to 2008. We pruned the network to the largest connected component (1,543 nodes and 15,483 edges). An author's role depends recursively on the number of connections to other authors, and the roles of those others. Hence, it measures collaboration. We use the G-index as a proxy measure for coauthor role (H-index provides similar results and thus is omitted here). The G-index measures the influence of a scientific author's publications, its value being the largest integer G such that the G most cited publications have at least G^2 citations. While G-index and coauthor role are not precisely the same, G-index score is influenced strongly by the underlying role. High-impact authors tend to be highly connected, especially with other high-impact authors. If a paper is highly cited, this boosts the score of every coauthor. Thus, we expect that if two authors have similar G-index scores, their node-pair is likely to have a high role similarity value. To normalize RoleSim, P-SimRank, and G-index values, we converted each raw value to a percentile rank.

Figure 6(a) addresses our second validation question (high rank→ similar roles?). For the top-ranked 0.01% of author-pairs, their difference in G-index ranking is about 20 points, for both RoleSim and P-SimRank, well below the random-pair value of 33. A below-average difference confirms that the authors are relatively similar. However, as we expand the search toward 10%, RoleSim continues to detect authors with similar authorship performance, whereas P-SimRank converges to random scoring.

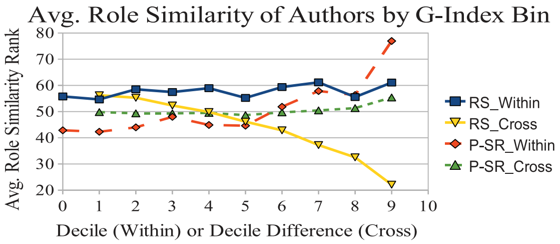


Fig. 7. Similarity of authors binned by K-index.

To validate *role* → *rank* performance, we binned the authors into 10 roles based on G-index value (bottom 10%, next 10%, etc.). For every pair of authors within the same role decile, we looked up role similarity percentile rank and computed an average per bin. We also computed averages for pairs of authors not in the same bin (dissimilar roles). Figure 7 shows our results. The average within-bin RoleSim value is consistently between 55% and 60%, better than the random-pair score of 50, and independent of whether the G-index is high or low. It performs equally well for all roles. P-SimRank within-bin scores (dashed line), however, are inconsistent. Performance of P-SimRank is worse than random for low G-scores, perhaps due to low density of links in the network. For the cross-bin data, the X-axis is the difference in decile bins for the two authors in a pair. The falling line of RoleSim indicates that role similarity correctly decreases as G-index scores become less similar. For P-SimRank, however, the cross-bin scores (dashed line) hover around 50, equivalent to random scoring.

6.4. Real Dataset: Internet Network

Our second dataset is a snapshot of the Internet at the level of autonomous systems (22,963 nodes and 48,436 edges), as generated by Newman [2006]. Several studies have confirmed that the Internet is hierarchically organized, with a densely connected core, medium density islands, a low-density connecting mesh, and stubs (singly connected nodes) at the periphery [Tauro et al. 2001; Carmi et al. 2007]. A node's position within the network (proximity to a hub or to a bridge between hubs) and its relation to others (such as density of connections) affects its efficiency for routing and its robustness. Proximity to the core is not a sufficient descriptor of role. For example, nodes that are in different local hubs might play the same role, even though they are far apart and may have somewhat different relationships to the main core. In Carmi et al. [2007], K-shell decomposition is shown to be an effective way to partition the graph into its hierarchical components, because it addresses how many connections and to whom. We use RoleSim to partition the nodes into role classes and compare this to the K-shell partitions.

The K-core of a graph is the induced subgraph where every node connects to at least K other nodes in the subgraph. If $K' > K$, then the K'-core must be an induced subgraph of the K-core. The K-shell is defined as the 'ring' of nodes that are included in a graph's $(K-1)$-core but not its K-core. In other words, we can decompose a graph into a set of nested rings, becoming denser as we move inward.

Using K-shells as our roles, we perform tests and analyses similar to those of the coauthor network. In Figure 6(b), we see that both measures do well for the top 0.1%, but P-SimRank's falters significantly when the range is expanded to the top 1%.

Next, we treat K-shells the same way that we treated G-index decile bins in the previous test (see Figure 8). Unlike decile bins, the shells do not have equal sizes. K-shells 1, 2, and 3 together contain 92% of all nodes. To clarify how these three shells dominate, we also show horizontal lines representing the combined weighted average

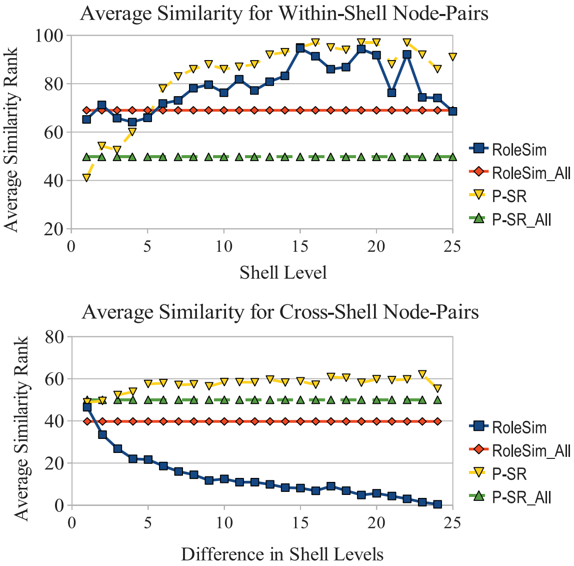


Fig. 8. Similarity of authors grouped by K-shell.

rank of all within-shell comparisons. RoleSim's within-shell values are consistently high, averaging 70%. Conversely, P-SimRank finds strong above-average similarity for the small high–K-shells but nearly random similarity for shells 1 through 3, pulling its overall performance down to 50%.

In cross-shell analysis, RoleSim is able to distinguish different shells very well: RoleSim approaches zero as shell difference approaches maximum. On the other hand, P-SimRank shows almost no correlation to shell difference. Many of its scores are above average when they should be below average (dissimilar). On the whole, it seems that P-SimRank is not detecting role, but something related to connectedness and density.

In all of these experiments, we can see that RoleSim provides a positive answer to the role similarity ranking: (1) node-pairs with similar roles have higher RoleSim ranking than node-pairs with dissimilar roles, and (2) high RoleSim ranking indicates that nodes have similar roles. P-SimRank scores, however, do not correlate with network role similarity.

6.5. Performance of Iceberg RoleSim

In this experiment, we study how Iceberg RoleSim performs in terms of reducing computational time and storage, and its accuracy at approximating the RoleSim score for high similar node-pairs. Here, we generated 12 scale-free graphs with up to $100K$ nodes and edge densities of 2 and 5, yielding up to $500K$ edges. We compared standard RoleSim to Iceberg RoleSim, with θ values of 0.8 and 0.9. The parameter α, which is the weighting for estimated nonstored values, is set to midpoint 0.5. For the scale-free graphs, the relative scale of the iceberg compared to the full similarity matrix depends on θ and edge density, but it is almost independent of the number of nodes. Table V shows that the icebergs' hash tables are only 0.15% to 3.5% of the full similarity matrices. Higher-density graphs tend to have more structural variation and thus fewer highly similar node pairs. In Figure 9, we see that Iceberg RoleSim is at least an order of magnitude faster. To check that the ranking has not changed significantly, we computed the Pearson correlation coefficient for each graph's Iceberg RoleSim's rankings versus the rankings from the corresponding portion of the full similarity matrix. For $\theta = 0.8$,

Table V. Iceberg Size Relative to RoleSim Matrix

Edge Density	Iceberg Size (as fraction of full matrix)	
$(\lvert E\rvert/\lvert V\rvert)$	$\theta = 0.8$	$\theta = 0.9$
1	2.77%	1.47%
2	2.47%	0.63%
5	3.53%	0.15%

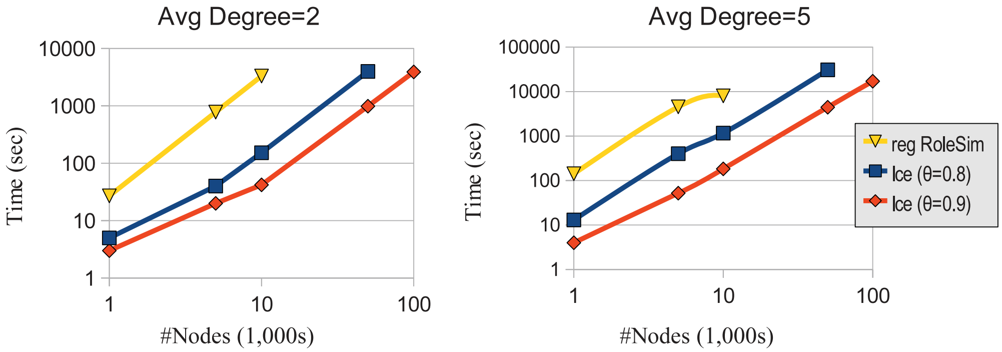


Fig. 9. Execution time: Standard versus Iceberg.

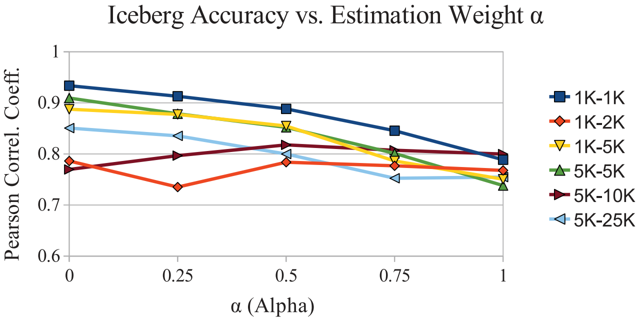


Fig. 10. Iceberg accuracy versus α.

the average coefficient is 0.823, and for $\theta = 0.9$, it is 0.880. Both show very strong correlation, indicating Iceberg-RoleSim's very good accuracy at ranking role similarity pairs.

Next we fixed θ at 0.9 and varied α from 0 to 1.0 to measure sensitivity of the accuracy of Iceberg RoleSim with respect to α. The results from six scale-free graphs are shown in Figure 10. The labels describe the number of nodes and edges of each graph. Most graphs prefer $\alpha = 0$, but some prefer a midrange value. Any value in the lower half seems acceptable.

6.6. Case Study: Coauthor Similarity

To better illustrate the role similarity ranking ability of RoleSim, we performed the following case study. We generated the coauthor network by taking all publications for SIGMOD, VLDB, ICDE, KDD, ICDM, and SDM from 2006 to 2011, as extracted from the DBLP database [Ley et al. 2012]. The resulting network contains 7,072 nodes with 18,994 edges. We computed the node similarity for all pairs of nodes using four algorithms (Iceberg RoleSim, SimRank, SimRank++, and PageSim). Then, for a given

Table VI. Top 10 Authors Similar to “Jiawei Han, G-index=162, H-index=76”

Iceberg RoleSim			SimRank++			PageSim		
Author	G	H	Author	G	H	Author	G	H
Philip S. Yu	112	63	Hyungsul Kim	0	0	Hong Cheng	31	17
Christos Faloutsos	134	67	Yuanyuan Zhou	54	32	Xifeng Yan	55	27
Haixun Wang	47	24	Gopal Krishna	4	2	Yizhou Sun	9	6
Jeffrey Xu Yu	41	28	Zhenmin Li	58	12	Hyungsul Kim	0	0
Jian Pei	89	37	Shobha Vasudevan	8	5	Bolin Ding	13	7
Beng Chin Ooi	56	35	Xiao Ma	16	8	Jing Gao	16	10
Divesh Srivastava	81	44	Tarek F. Abdelzaher	88	50	Chi Wang	6	5
Wei Fan	45	24	David Sheridan	4	2	Dong Xin	37	18
Anthony K. H. Tung	39	21	Kyuhyung Lee	2	2	Xiaolei Li	23	14
Samuel Madden	96	41	Joshua M. Hailpern	6	5	Philip S. Yu	112	63
Mean	74	38.4	Mean	24	11.8	Mean	30.2	16.7
Std. Dev.	33.35	15.99	Std. Dev.	31.01	16.34	Std. Dev.	33.08	18.04

Table VII. Top 10 Authors Similar to “Xifeng Yan, G-index=55, H-index=27”

Iceberg RoleSim			SimRank++			PageSim		
Author	G	H	Author	G	H	Author	G	H
Hong Cheng	31	17	Shu Tao	22	13	Shu Tao	22	13
Wei Fan	45	24	Arijit Khan	2	1	Hong Cheng	31	17
Charu C. Aggarwal	76	36	Supriyo Chakraborty	3	2	Ziyu Guan	5	3
Jimeng Sun	31	17	Nan Li	22	10	Nan Li	22	10
Hans-Peter Kriegel	111	45	Louise E. Moser	55	30	Nikos Anerousis	7	5
Jie Tang	45	26	Gengxin Miao	7	3	Xiaohui Gu	29	17
Xuemin Lin	37	22	Chen Chen	39	13	Arthur Gretton	38	19
Qiang Yang	56	36	Le Song	14	8	Le Song	14	8
Nick Koudas	65	35	Arthur Gretton	38	19	Marisa Thoma	0	0
Raghu Ramakrishnan	99	52	Marisa Thoma	0	0	Chen Chen	39	13
Mean	59.6	31	Mean	20.2	9.9	Mean	20.7	10.5
Std. Dev.	28.04	11.79	Std. Dev.	18.67	9.41	Std. Dev.	13.82	6.43

focal author, we found the 10 other authors that were most similar. We then used both G-index and H-index scores, obtained from the Microsoft Academic Search database [Microsoft Research 2012], as candidate measures of coauthorship role. If a particular similarity measure is a good role measure, then the 10 other authors should tend to have roles similar to that of the focal author, as measured by G- and H-indices. For example, we used Jaiwan Hei as our first focal author, who has a G-index score of 162. Our results are summarized in Table VI. Iceberg RoleSim found other high G-index authors, such as Philip Yu and Christos Faloutsos. All 10 had high G-index and H-index scores. SimRank and SimRank++, on the other hand, found mostly low-index authors. Since SimRank++ always performed better than SimRank, our tables show only the SimRank++ results. PageSim did a bit better. Its list includes a few strong authors, but most of the top 10 were also low G-index authors. We performed the same test for two other focal authors: Xifeng Yan, with G-index of 55 (Table VII), and Xiaolei Li, with G-index of 23 (Table VIII). Again, RoleSim was successful at finding authors with similar prestige, whereas the other algorithms were not.

7. CONCLUSION

We have developed RoleSim, the first real-valued role similarity measure that confirms automorphic equivalence. We have also presented a set of axioms that can test any

Table VIII. Top 10 Authors Similar to “Xiaolei Li, G-index=23, H-index=14”

Iceberg RoleSim			SimRank++			PageSim		
Author	G	H	Author	G	H	Author	G	H
Zhijun Yin	8	3	Tianyi Wu	8	5	Tianyi Wu	8	5
Yintao Yu	6	4	Jacob Lee	5	4	Zhijun Yin	8	3
Peixiang Zhao	9	4	Ricardo Redder	2	1	Hector Gonzalez	17	10
Qiaozhu Mei	24	14	Xiaoxin Yin	29	16	Jacob Lee	5	4
Jae-Gil Lee	16	7	John Paul Sondag	1	1	Ricardo Redder	2	1
Hector Gonzalez	17	10	Margaret Myslinska	1	1	Peixiang Zhao	9	4
Tianyi Wu	8	5	Peixiang Zhao	9	4	Yizhou Sun	9	6
Dong Xin	37	18	Zhijun Yin	8	3	Margaret Myslinska	1	1
Xin Jin	20	12	Lu An Tang	2	1	John Paul Sondag	1	1
Marina Danilevsky	1	1	Diego Klabjan	18	10	Dong Xin	37	18
Mean	14.6	7.8	Mean	8.3	4.6	Mean	9.7	5.3
Std. Dev.	10.56	5.49	Std. Dev.	8.94	4.88	Std. Dev.	10.74	5.25

future measure to see if it is an admissible measure or metric. Our experimental tests demonstrate RoleSim’s correctness and usefulness on real-world data, opening up exciting possibilities for scientific and business applications. At the same time, we see that other well-known measures, while suitable for other tasks, are not suitable for role similarity. This axiomatic approach may prove useful for developing and validating solutions to other related tasks.

APPENDIX

A. PROOFS OF THEOREMS AND LEMMAS

Proof for Theorem 3: Guaranteed Termination. Let $\delta^k(u, v)$ denote $RoleSim^k(u, v) - RoleSim^{k-1}(u, v)$, the difference of $RoleSim(u, v)$ scores between iterations k and $(k-1)$. In addition, let $D_k = \max_{(u,v)} |\delta^k(u, v)|$ be the maximum absolute difference across all u and v in iteration k. For any node-pair (u, v), let the maximal weighted matching between $N(u)$ and $N(v)$ computed at iteration $k+1$ be $\mathcal{M}^{k+1}$. Note that its weight $w(\mathcal{M}^{k+1}) = \sum_{(x,y)\in\mathcal{M}^{k+1}} RoleSim^k(x, y)$, a summation of $|\mathcal{M}| = min(d_u, d_v)$ terms. $|\mathcal{M}|$ is independent of k. Without loss of generality, assume $d_u \leq d_v$ so that $max(d_u, d_v) = d_v$ and $|\mathcal{M}| = d_u$. Given this, we observe that

$$\begin{aligned} w(\mathcal{M}^{k+1}) - (d_v \cdot D_k) &\leq w(\mathcal{M}^{k+1}) - |\mathcal{M}| \cdot D_k \\ &\leq w(\mathcal{M}^k) \\ &\leq w(\mathcal{M}^{k+1}) + |\mathcal{M}| \cdot D_k \\ &\leq w(\mathcal{M}^{k+1}) + (d_v \cdot D_k). \end{aligned}$$

Therefore, $|w(\mathcal{M}^{k+1}) - w(\mathcal{M}^k)| \leq d_v \times D_k$. Then,

$$\begin{aligned} |\delta^{k+1}(u, v)| &= |RoleSim^{k+1}(u, v) - RoleSim^k(u, v)| \\ &= |(1-\beta)\frac{w(\mathcal{M}^{k+1})}{d_v} - (1-\beta)\frac{w(\mathcal{M}^k)}{d_v}| \\ &= \frac{(1-\beta)}{d_v}|w(\mathcal{M}^{k+1}) - w(\mathcal{M}^k)| \\ &\leq \frac{(1-\beta)}{d_v} d_v \times D^k. \end{aligned}$$

Then $D^{k+1} = \max_{(u,v)} |\delta^{k+1}(u, v)| \leq (1-\beta)D^k$, so D^k decreases exponentially as k increases. Thus, $\lim_{k\to\infty} D_k = 0$ and $\lim_{k\to\infty} |RoleSim^k(u, v) - RoleSim^{k-1}(u, v)| < \delta$, for any $\delta > 0$. □

Proof for Lemma 4.3: Triangle Inequality Invariant. For iteration k, for any nodes a, b, and c, $d^k(a, c) \leq d^k(a, b) + d^k(b, c)$, where $d^k(a, b) = 1 - RoleSim^k(a, b)$. We must prove that this inequality still holds for the next iteration: $d^{k+1}(a, c) \leq d^{k+1}(a, b) + d^{k+1}(b, c)$. To facilitate our discussion, we abbreviate $RoleSim^k(u, v)$ as $r(u, v)$, and without loss of generality let $N_a \leq N_c$.

We utilize the following observation: *if there is a matching M between $N(a)$ and $N(c)$ that satisfies $1 - ((1-\beta)\frac{w(M)}{N_c} + \beta) \leq d^{k+1}(a, b) + d^{k+1}(b, c)$, then $d^{k+1}(a, c) \leq d^{k+1}(a, b) + d^{k+1}(b, c)$.* This is because $\frac{w(M)}{N_c} \leq \frac{w(\mathcal{M})}{N_c}$, where $\mathcal{M}$ is the maximal weighted matching between $N(a)$ and $N(c)$, and thus $1-((1-\beta)\frac{w(M)}{N_c}+\beta) \geq 1-((1-\beta)\frac{w(\mathcal{M})}{N_c}+\beta) = d^{k+1}(a, c)$.

In addition, we also denote the maximal weighted matching between $N(a)$ and $N(b)$ as $\mathcal{M}(a, b)$ and the maximal weighed matching between $N(b)$ and $N(c)$ as $\mathcal{M}(b, c)$. Now we consider three cases characterizing the relationship between $N(a)$, $N(b)$, and $N(c)$.

Case 1 ($N_b \leq N_a \leq N_c$): In this case, we observe $|\mathcal{M}(a, b)| = |\mathcal{M}(b, c)| = N_b$. Given this, we consider the following matching M between $N(a)$ and $N(c)$:

$$M = \{(x, z)|(x, y) \in \mathcal{M}(a, b) \wedge (y, z) \in \mathcal{M}(b, c)\},\ |M| = N_b.$$

Then, we have the following relationships:

$$\begin{aligned}
&d^{k+1}(a, b) + d^{k+1}(b, c) - \left(1 - (1-\beta)\frac{w(M)}{N_c} - \beta\right)\\
&= (1-\beta)\left[-\frac{w(\mathcal{M}(a, b))}{N_a} - \frac{w(\mathcal{M}(b, c))}{N_c} + \frac{w(M)}{N_c}\right] + 1 - \beta\\
&= (1-\beta)\left[\frac{N_b - w(\mathcal{M}(a, b))}{N_a} - \frac{N_b}{N_a} + \frac{N_b - w(\mathcal{M}(b, c))}{N_c} - \frac{N_b}{N_c}\right.\\
&\qquad \left. - \frac{N_b - w(M)}{N_c} + \frac{N_b}{N_c}\right] + 1 - \beta\\
&\geq (1-\beta)\left[1 - \frac{N_b}{N_a} + \frac{\sum_{(x,y)\in\mathcal{M}(a,b)}(1 - r(x, y))}{N_c}\right.\\
&\qquad \left. + \frac{\sum_{(y,z)\in\mathcal{M}(b,c)}(1 - r(y, z))}{N_c} - \frac{\sum_{(x,z)\in M}(1 - r(x, z))}{N_c}\right]\\
&\geq (1-\beta)\left[\frac{\sum_{(x,y,z)}(d^k(x, y) + d^k(y, z) - d^k(x, z))}{N_c}\right] \geq 0,
\end{aligned}$$

$$\text{where } (x, y) \in \mathcal{M}(a, b), (y, z) \in \mathcal{M}(b, c), (x, z) \in M.$$

Case 2 ($N_a \leq N_b \leq N_c$): In this case, we observe $|\mathcal{M}(a, b)| = N_a$ and $|\mathcal{M}(b, c)| = N_b$. It follows that there is a subset $n(b)$ of $N(b)$ of size N_a that participates in both $\mathcal{M}(a, b)$ and $\mathcal{M}(b, c)$: $n(b) = \{y|(y, z) \in \mathcal{M}(b, c)\backslash\{(y, z)|\ \nexists(x, y) \in \mathcal{M}(a, b)\}\}$. Given this, we consider the following matching M between $N(a)$ and $N(c)$:

$$M = \{(x, z)|(x, y) \in \mathcal{M}(a, b) \wedge (y, z) \in \mathcal{M}(b, c)\},$$

where $|M| = N_a$. Then, we have the following relationships:

$$\begin{aligned}
&d^{k+1}(a,b) + d^{k+1}(b,c) - \left(1 - (1-\beta)\frac{w(M)}{n_c} - \beta\right) \\
&= (1-\beta)\left[-\frac{w(\mathcal{M}(a,b))}{n_b} - \frac{w(\mathcal{M}(b,c))}{n_c} + \frac{w(M)}{n_c}\right] + 1 - \beta \\
&= (1-\beta)\left[\frac{n_a - w(\mathcal{M}(a,b))}{n_b} - \frac{n_a}{n_b} + \frac{n_a - w(\mathcal{M}(b,c))}{n_c} - \frac{n_a}{n_c}\right. \\
&\quad \left. - \frac{n_a - w(M)}{n_c} + \frac{n_a}{n_c}\right] + 1 - \beta \\
&\geq (1-\beta)\left[1 - \frac{n_a}{n_b} + \frac{\sum_{(x,y)\in\mathcal{M}(a,b)}(1 - r(x,y))}{n_c}\right. \\
&\quad + \frac{\sum_{(y,z)\in\mathcal{M}(b,c)\setminus\{(y,z)|\nexists(x,y)\in\mathcal{M}(a,b)\}}(1 - r(y,z))}{n_c} \\
&\quad \left. - \frac{n_b - n_a}{n_c} - \frac{\sum_{(x,z)\in M}(1 - r(x,z))}{n_c}\right] \\
&\geq (1-\beta)\left[1 - \frac{n_a}{n_b} - \frac{n_b - n_a}{n_c}\right. \\
&\quad \left. + \frac{\sum_{(x,y,z)}(d^k(x,y) + d^k(y,z) - d^k(x,z))}{n_c}\right], \\
&\text{where } (x,y) \in \mathcal{M}(a,b), (y,z) \in \mathcal{M}(b,c), (x,z) \in M
\end{aligned}$$

$$\begin{aligned}
&\geq (1-\beta)\left[1 - \frac{n_a}{n_b} - \frac{n_b}{n_c} + \frac{n_a}{n_c}\right] \\
&= (1-\beta)\frac{n_b n_c - n_a n_c - n_b^2 + n_a n_b}{n_b n_c} \\
&= (1-\beta)\frac{(n_b - n_a)(n_c - n_b)}{n_b n_c} \\
&\geq 0
\end{aligned}$$

Case 3 ($N_a \leq N_c \leq N_b$): In this case, we observe $|\mathcal{M}(a,b)| = N_a$ and $|\mathcal{M}(b,c)| = N_c$. Given this, we consider the following matching M between $N(a)$ and $N(c)$:

$$M = \{(x,z)|(x,y) \in \mathcal{M}(a,b) \wedge (y,z) \in \mathcal{M}(b,c)\}.$$

In addition, we define

$$\begin{aligned}
M_1 &= \{(x,y)|(x,y) \in \mathcal{M}(a,b) \wedge \nexists(y,z) \in \mathcal{M}(b,c)\}. \\
M_2 &= \{(y,z)|(y,z) \in \mathcal{M}(b,c) \wedge \nexists(x,y) \in \mathcal{M}(a,b)\}.
\end{aligned}$$

In other words, $M_1 \subset \mathcal{M}(a,b)$ and $M_2 \subset \mathcal{M}(b,c)$ do not link to each other using intermediate node $y \in N(b)$. We further denote $m_1 = |M_1|$, $m_2 = |M_2|$, $m_3 = |M|$. Note that $m_1 = N_a - m_3$, $m_2 = N_c - m_3$, and $N_b \geq m_1 + m_2 + m_3$.

Then, we have the following relationships:

$$
\begin{aligned}
&d^{k+1}(a,b)+d^{k+1}(b,c)-\left(1-(1-\beta)\frac{w(M)}{N_c}-\beta\right)\\
&\quad\geq d^{k+1}(a,b)+d^{k+1}(b,c)-\left(1-(1-\beta)\frac{w(M)}{N_b}-\beta\right)\\
&\quad\geq 1-\beta-(1-\beta)\left(\frac{w(\mathcal{M}(a,b))}{N_b}+\frac{w(\mathcal{M}(b,c))}{N_b}-\frac{w(M)}{N_b}\right)\\
&\quad=(1-\beta)\left(1+\frac{m_3-w(\mathcal{M}(a,b))}{N_b}-\frac{m_3}{N_b}+\frac{m_3-w(\mathcal{M}(b,c))}{N_b}-\frac{m_3}{N_b}-\frac{m_3-w(M)}{N_b}+\frac{m_3}{N_b}\right)\\
&\quad\geq(1-\beta)\left(1-\frac{m_3}{N_b}+\frac{\sum_{(x,y)\in\mathcal{M}(a,b)\setminus M_1}(1-r(x,y))}{N_b}-\frac{m_1}{N_b}\right.\\
&\qquad\left.+\frac{\sum_{(y,z)\in\mathcal{M}(b,c)\setminus M_2}(1-r(y,z))}{N_b}-\frac{m_2}{N_b}-\frac{\sum_{(x,z)\in M}(1-r(x,z))}{N_b}\right)\\
&\quad\geq(1-\beta)\left(1-\frac{m_3}{N_b}-\frac{m_1}{N_b}-\frac{m_2}{N_b}+\frac{\sum_{(x,y,z)}(d^k(x,y)+d^k(y,z)-d^k(x,z))}{N_b}\right)\\
&\quad\geq((x,y)\in\mathcal{M}(a,b),(y,z)\in\mathcal{M}(b,c),(x,z)\in M)\left(1-\beta)(1-\frac{m_1+m_2+m_3}{N_b}\right)\geq 0\qquad\square
\end{aligned}
$$

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under CAREER grant no. IIS-0953950, by the National Institute of General Medical Science of the National Institutes of Health (NIGMS/NIH) under Award no. R01GM103309 and by the Ohio Supercomputer Center under Grant no. PGS0218.

REFERENCES

Ioannis Antonellis, Hector Garcia-Molina, and Chi-Chao Chang. 2008. Simrank++: Query rewriting through link analysis of the clickgraph. *Proc. VLDB Endow.* 1, 1, 408–421.

D. Avis. 1983. A survey of heuristics for the weighted matching problem. *Network* 13, 475–493.

Vladimir Batagelj, Patrick Doreian, and Anuška Ferligoj. 1992. An optimizational approach to regular equivalence. *Social Networks* 14, 121–135.

Stephen P. Borgatti and Martin G. Everett. 1992. Notions of position in social network analysis. *Sociological Methodology* 22, 1–35.

Stephen P. Borgatti and Martin G. Everett. 1993. Two algorithms for computing regular equivalence. *Social Networks* 15, 361–376.

Yuanzhe Cai, Gao Cong, Xu Jia, Hongyan Liu, Jun He, Jiaheng Lu, and Xiaoyong Du. 2009. Efficient algorithm for computing link-based similarity in real world networks. In *Ninth IEEE Int. Conf. Data Mining (ICDM)*. IEEE Computer Society, 734–739.

Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. 2007. A model of Internet topology using k-shell decomposition. *In Proc. Nat'l Academy Sci. (PNAS)* 104, 27, 11150–11154.

Dragos M. Cvetkovíc, Michael Doob, and Horst Sachs. 1998. *Spectra of Graphs: Theory and Applications, 3rd Revised and Enlarged Edition*. Wiley.

Patrick Doreian, Vladimir Batagelj, and Anuška Ferligoj. 2005. *Generalized Blockmodeling*. Vol. 25. Cambridge University Press.

Natalia Dragan, Michael L. Collard, and Jonathan I. Maletic. 2009. Using method stereotype distribution as a signature descriptor for software systems. In *IEEE Int. Conf. Software Maintenance (ICSM)*. IEEE, 567–570.

Martin G. Everett and Stephen P. Borgatti. 1996. Exact colorations of graphs and digraphs. *Social Networks* 18, 319–331.

Dániel Fogaras and Balázs Rácz. 2005. Scaling link-based similarity search. In *Proc. 14th Int. Conf. World Wide Web (WWW)*. ACM, 641–650.

Scott Fortin. 1996. *The Graph Isomorphism Problem*. Technical Report TR 96-20. Dept. Computer Science, University of Alberta, Edmonton, Alberta, Canada.

Linton C. Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry* 40, 1, 35–41.

Chris Godsil and Gordon Royle. 2001. *Algebraic Graph Theory*. Springer-Verlag.

Emilie M. Hafner-Burton, Miles Kahler, and Alexander H. Montgomery. 2009. Network analysis for international relations. *International Organization* 63, 3, 559–592.

Petter Holme and Mikael Huss. 2005. Role-similarity based functional prediction in networked systems: Application to the yeast proteome. *J. R. Soc. Interface* 2, 4, 327–333.

Glen Jeh and Jennifer Widom. 2002. SimRank: A measure of structural-context similarity. In *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining (KDD)*. ACM, 538–543.

Xu Jia, Yuanzhe Cai, Hongyan Liu, Jun He, and Xiaoyong Du. 2009. Calculating similarity efficiently in a small world. In *Proc. 5th Int. Conf. Advanced Data Mining Applications (ADMA)*. Springer-Verlag, Berlin, Heidelberg, 175–187. DOI:http://dx.doi.org/10.1007/978-3-642-03348-3_19

Ruoming Jin, Victor E. Lee, and Hui Hong. 2011. Axiomatic ranking of network role similarity. In *KDD*. ACM, 922–930.

M. M. Kessler. 1963. Bibliographic coupling between scientific papers. *American Documentation* 14, 1, 10–25.

H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1–2, 83–97.

Victor E. Lee, Ning Ruan, Ruoming Jin, and Charu Aggarwal. 2010. *Managing and Mining Graph Data*. Springer, Chapter 10: A survey of algorithms for dense subgraph discovery, 303–336.

E. A. Leicht, Petter Holme, and Mark E. J. Newman. 2005. Vertex similarity in networks. *Phys. Rev. E* 73, 2, 026120.

Michael Ley, Marc Herbstritt, Marcel R. Ackermann, Oliver Hoffmann, Michael Wagner, Stefanie von Keutz, Katharina Hostert, and Doris Holzträger. 2012. *The DBLP Computer Science Bibliography*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. http://www.informatik.uni-trier.de/∼ley/db/.

Pei Li, Yuanzhe Cai, Hongyan Liu, Jun He, and Xiaoyong Du. 2009. Exploiting the block structure of link graph for efficient similarity computation. In *Proc. 13th Pacific-Asia Conf. Advances Knowledge Discovery Data Mining (PAKDD)*. Springer-Verlag, Berlin, Heidelberg, 389–400. DOI:http://dx.doi.org/10.1007/978-3-642-01307-2_36

Zhenjiang Lin, Irwin King, and Michael R. Lyu. 2006. PageSim: A novel link-based similarity measure for the World Wide Web. In *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence*. IEEE Computer Society, 687–693.

Zhenjiang Lin, Michael R. Lyu, and Irwin King. 2007. Extending link-based algorithms for similar Web pages with neighborhood structure. In *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence*. IEEE Computer Society, 263–266. http://www.cse.cuhk.edu.hk/∼king/PUB/WI2007_Lin.pdf.

Zhenjiang Lin, Michael R. Lyu, and Irwin King. 2009. MatchSim: A novel neighbor-based similarity measure with maximum neighborhood matching. In *Proc. 18th ACM Conf. Inform. Knowledge Manage. (CIKM)*. ACM, 1613–1616.

Dmitry Lizorkin, Pavel Velikhov, Maxim Grinev, and Denis Turdakov. 2008. Accuracy estimate and optimization techniques for SimRank computation. In *Proc. VLDB Endow.* 1, 1, 422–433. DOI:http://dx.doi.org/10.1145/1453856.1453904

F. P. Lorrain and H. C. White. 1971. Structural equivalence of individuals in networks. *J. Math. Sociology* 1, 49–80.

J. J. Luczkovich, Stephen P. Borgatti, J. C. Johnson, and Martin G. Everett. 2003. Defining and measuring trophic role similarity in food webs using regular coloration. *J. Theoretical Biology* 220, 3, 303–321.

Ben D. MacArthur, Rubén J. Sánchez-García, and James W. Anderson. 2008. Note: Symmetry in complex networks. *J. Discrete Applied Math.* 156, 18, 3525–3531.

Maarten Marx and Michael Masuch. 2003. Regular equivalence and dynamic logic. *Social Networks* 25, 1, 51–65.

B. D. McKay. 1981. Practical graph isomorphism. *Congressus Numerantium* 30, 45–87.

Guy Melançon and Arnaud Sallaberry. 2008. Edge metrics for visual graph analytics: A comparative study. In *Proc. 12th Int. Conf. Inform. Visual.* IEEE Computer Society, 610–615. DOI:http://dx.doi.org/10.1109/IV.2008.10

Microsoft Research. 2012. Microsoft academic search. http://academic.research.microsoft.com/RankList?entitytype=2&topdomainid=2&subdomainid=7. (2012). Accessed August 2012.

Mark Newman. 2006. Internet network. http://www-personal.umich.edu/∼mejn/netdata/.

Mark E. J. Newman. 2004. Coauthorship networks and patterns of scientific collaboration. In *Proc. Nat'l Academy Sci. (PNAS)* 101, Suppl 1, 5200–5205.

Evelien Otte and Ronald Rousseau. 2002. Social network analysis: A powerful strategy, also for the information sciences. *J. Information Science* 28, 6, 441–453.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.

Ronald Read and Derek Corneil. 1977. The graph isomorphism disease. *J. Graph Theory* 1, 339–363.

Michael Schultz and Mark Liberman. 1999. Topic detection and tracking using idf-weighted cosine coefficient. In *Proc. DARPA Broadcast News Workshop*. Morgan Kaufmann, 189–192.

Henry Small. 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Amer. Soc. Information Sci.* 24, 265–269.

Malcolm K. Sparrow. 1993. A linear algorithm for computing automorphic equivalence classes: The numerical signatures approach. *Social Networks* 15, 2, 151–170. DOI:http://dx.doi.org/10.1016/0378-8733(93)90003-4

Jie Tang, Jing Zhang, Limin Yao, and Juanzi Li. 2008. Extraction and mining of an academic social network. In *Proc. 17th Int. Conf. World Wide Web (WWW)*. ACM, 1193–1194. DOI:http://dx.doi.org/10.1145/1367497.1367722

T. T. Tanimoto. 1958. An elementary mathematical theory of classification and prediction. *IBM Taxonomy Application M. A.* 6, 3.

Sudhir L. Tauro, Georgos Siganos, C. Palmer, and Michalis Faloutsos. 2001. A simple conceptual model for the Internet topology. In *Proc. IEEE Global Telecomm. Conf.* IEEE, 1667–1671.

Yuchung J. Wang and George Y. Wong. 1987. Stochastic blockmodels for directed graphs. *J. American Statistical Assoc.* 82, 397, 8–19.

Stanley Wasserman and Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Douglas R. White and Karl P. Reitz. 1983. Graph and semigroup homomorphisms on networks of relations. *Social Networks* 5, 193–234.

Harrison White, Scott Boorman, and Ronald Breiger. 1976. Social structure from multiple networks. I: Blockmodels of roles and positions. *Am. J. Sociology* 81, 730–780.

Wensi Xi, Edward A. Fox, Weiguo Fan, Benyu Zhang, Zheng Chen, Jun Yan, and Dong Zhuang. 2005. SimFusion: Measuring similarity using unified relationship matrix. In *Proc. 28th Int. ACM SIG Conf. Research Develop. Inform. Retrieval (SIGIR)*. ACM, 130–137.

Erjia Yan and Ying Ding. 2009. Applying centrality measures to impact analysis: A coauthorship network analysis. *J. Am. Soc. Information Sci. Technology* 60, 10, 2107–2118.

Xiaoxin Yin, Jiawei Han, and Philip S. Yu. 2006. LinkClus: Efficient clustering via heterogeneous semantic links. In *Proc. 32nd Int. Conf. Very Large Data Bases (VLDB)*. VLDB Endowment, 427–438.

Peixiang Zhao, Jiawei Han, and Yizhou Sun. 2009. P-Rank: A comprehensive structural similarity measure over information networks. In *Proc. 18th ACM Conf. Inform. Knowledge Manage. (CIKM)*. ACM, 553–562.

Received September 2012; revised April 2013; accepted August 2013