Studies in Structural Similarity Measures over Information Networks

A Project Report

submitted by

SAI KIRAN NARAYANASWAMI

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY
DUAL DEGREE

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Studies in Structural Similarity Measures over Information Networks**, submitted by **N Sai Kiran**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology Dual Degree**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Venkatesh Ramaiyan
Research Advisor
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Dr. Balaraman Ravindran
Research Advisor
Associate Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

First and foremost, I thank my advisors Prof. Balaraman Ravindran and Prof. Venkatesh Ramaiyan for always being there to provide guidance despite their busy schedules, not to mention being tremendous sources of insight at the Networks Group meetings. I would also like to thank the other members of the group for the various discussions that provided new perspectives and exposure to other areas of research.

I thank my closest friends Santhosh, Nikhilesh and Sabyasachi (all DD EE '17) for being with me through thick and thin. I also thank IITM itself for making me who I am today.

Last but not the least, I thank my parents, without whose unwavering support I would be nothing.

ABSTRACT

Measuring similarity between nodes is a fundamental problem in the analysis of information networks, and also plays a key role in collaborative filtering and information retrieval tasks on networks such as web graphs and bibliographic networks. SimRank is a widely studied link-based similarity measure that is known for its simple, yet powerful philosophy that two nodes are similar if they are referenced by similar nodes. While this philosophy has been the basis of several improvements, there is another useful, albeit less frequently discussed interpretation for SimRank known as the Random Surfer-Pair Model. This interpretation has enabled SimRank to be used in a Monte Carlo framework, and has also led to the formulation of PSimRank which remediates a deficiency in SimRank. In this work, we show that other well known measures derived from SimRank can also be reinterpreted using Random Surfer-Pair Models, thus establishing them as a general and unifying framework for various link-based similarity measures. This also serves to provide new insights into their functioning and allows for using these measures in a Monte Carlo framework, which can potentially allow them to scale to very large graphs. We also develop a new measure, PSimRank* under this interpretation and demonstrate its effectiveness, thus opening up numerous possibilities for further developments. Finally, we investigate possibilities for handling negative examples in systems that use network data to provide recommendations. We propose an algorithm, PacRank as an attempt to utilize structural similarity measures to this end.

Contents

ACKNOWLEDGEMENTS i

ABSTRACT ii

LIST OF TABLES vi

LIST OF FIGURES vii

NOTATION viii

1 INTRODUCTION 1

1.1 Information Networks 1

1.2 Formalizing INs 2

1.3 Structural Similarity Measures 2

1.4 Contributions 4

1.5 Overview 4

2 STRUCTURAL SIMILARITY MEASURES 6

2.1 SimRank 6

2.1.1 Definition 6

2.1.2 Computing SimRank 7

2.1.3 Variants of SimRank 8

2.1.4 Deficiencies in SimRank 8

2.2 P-Rank 9

2.3 PSimRank 10

2.3.1 The Pairwise normalization problem 10

2.3.2 A solution : PSimRank 10

2.4 C-Rank 11

2.5 SimRank* 12

2.6 MatchSim 13

2.7 CoSimRank 13

3 Generalized Random Surfer-Pair Models **15**

3.1 Existing Random Surfer-Pair Models 15

3.1.1 SimRank 15

3.1.2 PSimRank 16

3.2 Generalizing the Random Surfer-Pair Model 17

3.3 Equivalence to recursive form 18

3.4 Reinterpreting Existing Measures 20

3.4.1 SimRank 20

3.4.2 PSimRank 20

3.4.3 C-Rank 21

3.4.4 P-Rank 21

3.4.5 SimRank* 21

3.5 Monte Carlo Computation 22

3.6 P+Rank : A more consistent measure 23

3.7 PSimRank* : Combining the Best of Both Worlds 25

3.8 Experiments 26

3.9 Other Possibilities : 28

3.9.1 Incorporating edge weights : 28

3.9.2 Termination and Scoring : 28

3.10 Benefits of the Generalized Random Surfer-Pair Model 29

3.10.1 Unification and Generalization 29

3.10.2 New Insights Into Existing Work 29

3.10.3 Computational Advantages 29

3.11 Conclusions : 30

4 Negative Examples in Recommendations **32**

4.1 Possibilities for handling negative examples with link based measures 32

4.2 PacRank 33

4.3 Incremental Feedback Based Querying : 34

4.4 Conclusions 35

A Proofs of Theorems **36**

A.1 Theorem 5 36
A.1.1 Monotonicity : 36
A.1.2 Boundedness : 36
A.1.3 Convergence : 37
A.2 Theorem 6 37
A.2.1 Existence 37
A.2.2 Uniqueness 38
A.3 Theorem 7 39

List of Tables

3.1 Left : MAP values attained for various values of λ by P-Rank and P+Rank. Right : B

List of Figures

1.1 A toy example network. 3

2.1 Illustration for the pairwise normalization problem. Source : Fogaras and Rácz (200

2.2 An example of the level-wise computation problem: There are no paths of equal len

3.1 Example of P-Rank ignoring paths because only one of them changes direction : P-R

NOTATION

$I_i(a)$	i^{th} in-neighbor of node a
$O_i(a)$	i^{th} out-neighbor of node a
$L_i(a)$	i^{th} undirected-neighbor of node a
$\mathbf{Q}$	Column normalized adjacency matrix
$s\,(a, b)$	Similarity of nodes a and b under the measure in context

Chapter 1

INTRODUCTION

1.1 Information Networks

Numerous situations arise in physical, social, electronic and other systems that involve many different entities interacting with each other in different ways. These interactions form vast networks with rich structures. These networks have been termed *Information networks* (INs) by Zhao *et al.* (2009). They are ubiquitous, with some well known examples being social networks, the World Wide Web, citation and collaboration networks (bibliometrics), and biological networks.

As advances in computing and information systems have enabled the collection of data on such networks on a large scale, the analysis of these networks is now of paramount importance and yields far reaching benefits. For example, analysis of transportation networks can be used to discover knowledge leading to more efficient operation (Jiang *et al.*, 2005). The study of biological interaction networks has been very useful in the study of several kinds of biological systems (Russell and Aloy, 2008).

It comes as no surprise then, that the information sciences can also benefit from such studies, one famous work in this field being PageRank (Brin and Page, 1998) that revolutionized information retrieval for the web. Social networks have also received tremendous attention owing to the rise of social media platforms. Another important application is in bibliometrics, which is concerned with networks formed by academic publications and collaboration such as citation and co-authorship networks.

1.2 Formalizing INs

Graphs are a natural way to represent a collection of entities interacting with or related to each other in some manner. There are many kinds of graphs that can model different kinds of data. A Web graph for instance would be directed, with edges indicating that one page links to another. A roadways network would have weights associated with the edges indicating the length of each road. There even exist other sophisticated models like hypergraphs used for more complex data.

This work deals with INs represented by directed graphs with unweighted edges, where all nodes are of the same kind. Zhao *et al.* (2009) refer to such INs as *homogeneous INs*. Henceforth, the terms "network" and "graph" are used interchangeably to refer to such an IN unless otherwise stated.

The graph is denoted by $G = (V, E)$, where V is the vertex set and E is the edge set, with elements of the form (a, b) indicating that there is an edge between nodes a and b. Each node a has associated with it two ordered sets : $I(a)$, the set of in-neighbors and $O(a)$, the set of out-neighbors. They can be indexed with an index i as $I_i(a)$ and $O_i(a)$. Sometimes the set of both in and out neighbors together, which we denote by $L(a)$ is also used. We denote by $\mathbf{Q}$ the column normalized adjacency matrix associated with the graph whose entries $\mathbf{Q}_{i,j} = 1/|I(i)|$ if there is an edge from j to i, and 0 otherwise. An example network is shown in Figure 1.1.

1.3 Structural Similarity Measures

A sizeable amount of data these days is available in the form of networks. Naturally, the need arises to effectively utilize this data in the relevant domains. One situation where network data proves very useful is in recommender systems. With the advent of electronic commerce, data is frequently available for products that customers have purchased. The utilization of this data to provide recommendations is known as collaborative filtering.

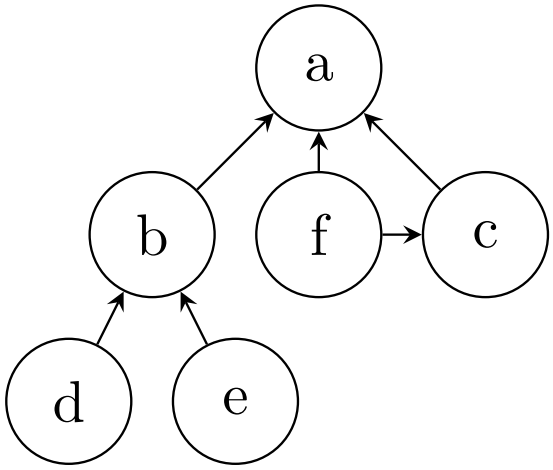

Figure 1.1: A toy example network.

As human knowledge grows, leading to rapidly expanding bodies of literature, there is an increasing requirement for effective recommender systems to aid researchers. Thus, utilizing bibliographic network data such as citation and co-authorship networks to provide recommendations is of rising importance.

Some form of similarity assessment would of course be a key component of any information retrieval system. This is specially true of recommender systems. Obviously, methods that can make use of vast network data would be at an advantage. In this work, we consider *structural similarity measures*, which work with only the link structure of the network. They are also known as *link-based similarity measures*. Although many of these were developed in the context of citation networks, they still apply to any directed networks.

Among the first such well known measures are Co-Citation (Small (1973)) and its counterpart Bibliographic coupling (Kessler (1963)) that are respectively the frequency with which nodes refer to two given nodes , and the frequency with which nodes are referenced together by two given nodes. Amsler (Amsler (1972)) is a combination of the former two measures.

SimRank (Jeh and Widom (2002)) was a seminal work that dramatically changed

the landscape of structural similarity. Its elegant and intuitive philosophy has been the basis of several future works. SimRank and methods derived from it have been successfully used in a variety of applications including natural language processing (Rothe and Schütze (2014)), clustering (Yin *et al.* (2006)) and even search query rewriting (Antonellis *et al.* (2008)).

There have also been similarity measures based on PageRank. Two notable examples of this are CoSimRank (Rothe and Schütze (2014)), which computes scores based on Personalized PageRank (Haveliwala (2002)), and PageSim (Lin *et al.* (2006)), which is based on propagation of PageRank scores.

1.4 Contributions

At the focus of this work are the similarity measures related to SimRank, particularly the probabilistic interpretation known as the Random Surfer-Pair model. A Generalized version of the Random Surfer-Pair model is proposed which extends this interpretation to several other existing structural similarity measures. This interpretation is used to develop a new measure which is shown to perform better than many other measures. The various benefits of such a model and and the possibilities for further developments are highlighted.

Next, the problem of incorporating negative examples in recommender systems is investigated. Some possible ways to adapt structural similarity measures for this purpose are explored. The PacRank algorithm is proposed as one way to accomplish this.

1.5 Overview

Chapter 2 surveys various structural similarity measures starting with SimRank. Section 2.1 presents the details of SimRank, followed by a discussion on its computation. Section 2.1.4 presents the main theoretical issues identified in SimRank. The remainder of the chapter is devoted to presenting several

other measures and how they address these issues in SimRank.

In Chapter 3, the original Random Surfer-Pair formulation is introduced in Section 3.1. Then the Generalized Random Surfer-Pair model is developed in Sections 3.2 and 3.3. The application of the model to several existing measures is demonstrated in Section 3.4. Section 3.5 discusses the use of Monte Carlo methods with the Generalized Random Surfer-Pair model. A theoretical deficiency of an existing measure, P-Rank under the new model is discussed in Section 3.6, followed by the corrected measure P+Rank. Next, a new measure, PSimRank* is proposed in Section 3.7. Experiments are performed in Section 3.8 to assess the performance of P+Rank and PSimRank. The chapter concludes after a summary of the advantages of the new formulation in Section 3.10.

Chapter 4 presents the problem of incorporating negative examples in recommender systems, and moves on to some attempts to utilize similarity measures for this purpose. The PacRank algorithm is proposed in Section 4.2, followed by ways to efficiently process queries with this algorithm in the practical setting of a paper recommender system in Section 4.3.

Chapter 2

STRUCTURAL SIMILARITY MEASURES

In this chapter, we begin with a discussion on SimRank, and highlight the major issues present in SimRank. Then, we discuss several other measures and how they solve these issues. Throughout this work, the notation $s(a, b)$ is used to denote the similarity of nodes a and b under the similarity measure being discussed in the context unless stated otherwise.

2.1 SimRank

2.1.1 Definition

SimRank is known for its simple, yet powerful philosophy :

> Two nodes are similar if they are referenced by similar nodes.

which leads to the recursive form that is most commonly used. Formally, the SimRank score of a pair of nodes a and b is the average of the SimRank scores of their in-neighbors taken pairwise, and damped by a constant $C < 1$:

$$s(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } |I(a)| = 0 \text{ or } |I(b)| = 0 \\ \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s\left(I_i(a), I_j(b)\right) & \text{otherwise} \end{cases} \tag{2.1}$$

The base cases for the recursion are that a node is maximally similar to itself, and that SimRank is taken to be zero (unavailable) when either node has no in-neighbors.

It is also possible to express the above definition in matrix form (Rothe and Schütze (2014); Yu *et al.* (2013)) :

$$\mathbf{S} = C \cdot \left(\mathbf{Q}^T \cdot \mathbf{S} \cdot \mathbf{Q}\right) + (1 - C) \cdot \mathbf{I} \tag{2.2}$$

2.1.2 Computing SimRank

The recursive definition lends itself to a natural iterative algorithm whose iterations are as follows :

$$s_{k+1}(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } |I(a)| = 0 \text{ or } |I(b)| = 0 \\ \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s_k\left(I_i(a), I_j(b)\right) & \text{otherwise} \end{cases} \tag{2.3}$$

With the initial conditions set according to the base cases as discussed above :

$$s_0(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases} \tag{2.4}$$

It is shown that the recursive form in equation 2.1 has a unique solution and that the above iterative process converges in the limit to this unique solution. The iterations are also monotonically increasing, that is $s_{k+1}(a,b) \geq s_k(a,b)$ for all node pairs (a,b).

Applying these iterations as such gives a time complexity of $O(N^4)$ in the worst case for computing similarity between all pairs of nodes. Radius based pruning is suggested in Jeh and Widom (2002) where similarities of nodes that are more than a particular distance (usually 2 or 3) apart (i.e outside a particular radius of each node) are set to zero as they are unlikely to be similar.

A dynamic programming based approach was proposed in Lizorkin *et al.* (2010) that reduces the complexity to $O(N^3)$. This approach involves memoization of partial sums in a manner similar to dynamic programming. Other optimizations such as selecting only "essential" node pairs and threshold sieving in order to avoid computing scores between pairs that are close to zero to further reduce computation are also presented in the same paper.

2.1.3 Variants of SimRank

A bipartite version is presented in Jeh and Widom (2002) that extends the intuition of SimRank to domains where nodes are of two different types and edges represent relationships between nodes of different classes. An example of such a situation would be people and items purchased by people. Here, the similarity between two nodes of one class is determined by how much they are referenced by or refer to similar nodes of the other class. This can also be applied to homogeneous domains to get a similarity analogue of HITS scores (Kleinberg (1999)).

Reverse SimRank, denoted as rvs-SimRank is a version of SimRank that uses out-links instead of in-links.

2.1.4 Deficiencies in SimRank

There have been numerous efforts to improve SimRank ever since its introduction. New algorithms that have been proposed try to rectify theoretical deficiencies in SimRank. Hamedani and Kim (2016) identify three main problems that most improvements to SimRank try to solve :

- The In Links Consideration Problem : SimRank is unavailable when either node has no in-neighbors, even though there may be evidence of similarity in the out-neighbors.
- The Pairwise Normalization Problem : This is the counter-intuitive effect that the SimRank score of a pair of nodes can *decrease* as there are more and more nodes referring to both of them. This will be discussed in more detail in Section 2.3.1.

- The Level-wise computation problem : SimRank is unavailable for node pairs that don't have any paths of *equal* length to a common node.

These issues will be discussed in detail alongside existing works that solve them in the upcoming sections.

2.2 P-Rank

P-Rank (Zhao *et al.* (2009)) was proposed to take into account out-links as well in computing similarity. It has the following recursive form :

$$s(a,b) = \lambda \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s\left(I_i(a), I_j(b)\right) + (1-\lambda) \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} s\left(O_i(a), O_j(b)\right) \quad (2.5)$$

It essentially adds an additional clause :

two entities are also similar if they reference similar entities

The base cases are similar to those of SimRank, except that only the term that corresponds to in (out) neighbors gets zeroed out if one or both of (a, b) doesn't have in (out) neighbors. This ensures that P-Rank is not unavailable for node pairs without in-links as is the case with SimRank, as long as they have out links.

A similar iterative procedure can be written, with the same convergence guarantees as for SimRank. The partial sums memoization approach of Lizorkin *et al.* (2010) can also be directly used to provide the same complexity reduction to $O\left(N^3\right)$.

λ is a tunable parameter that controls how much preference to give to in-links and out-links. SimRank and rvs-SimRank are in fact special cases when $\lambda = 1$

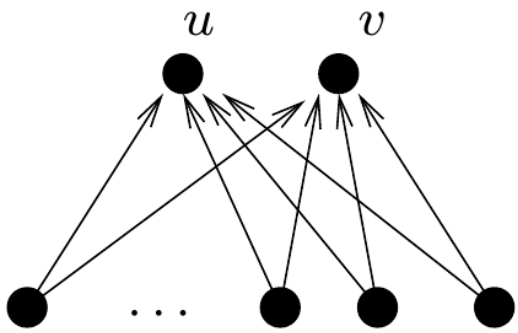

Figure 2.1: Illustration for the pairwise normalization problem. Source : Fogaras and Rácz (2005)

and $\lambda = 0$ respectively.

Co-citation, Bibliographic Coupling and Amsler can also be achieved as special cases when considering only one iteration of P-Rank.

2.3 PSimRank

2.3.1 The Pairwise normalization problem

Consider the situation shown in Figure 2.1. Two nodes u and v have several (say k) nodes that cite both of them. Now, if these nodes are unrelated to each other (i.e they have zero similarity), the SimRank score between u and v is found to be $\frac{C}{k}$, which *decreases* with k. Thus, even though there are more witnesses to the similarity of u and v, their SimRank score is reduced. This is known as the pairwise normalization problem.

2.3.2 A solution : PSimRank

The recursive form of PSimRank (Fogaras and Rácz (2005)) is as follows :

$$
\begin{aligned}
s\left(a,b\right) =& \frac{C|I\left(a\right)\cap I\left(b\right)|}{|I\left(a\right)\cup I\left(b\right)|} \\
&+\frac{C}{|I\left(a\right)\cup I\left(b\right)||I\left(b\right)|}\sum_{\substack{a^{'}\in I(a)\setminus I(b)\\ b^{'}\in I(b)}} s\left(a^{'},b^{'}\right) \\
&+\frac{C}{|I\left(a\right)\cup I\left(b\right)||I\left(a\right)|}\sum_{\substack{b^{'}\in I(b)\setminus I(a)\\ a^{'}\in I(a)}} s\left(a^{'},b^{'}\right)
\end{aligned}
\tag{2.6}
$$

PSimRank solves the pairwise normalization problem by assigning greater importance to common in-neighbors. This is done via the Jaccard coefficient of the sets of in-neighbors. The properties of PSimRank will be revisited under a different context subsequently.

2.4 C-Rank

C-Rank (Yoon *et al.* (2016)) is a version of PSimRank that operates on the underlying undirected network in order to consider out-links as well as solve the pairwise normalization problem like PSimRank. The recursive form is :

$$
\begin{aligned}
s\left(a,b\right) =& \frac{C|L\left(a\right)\cap L\left(b\right)|}{|L\left(a\right)\cup L\left(b\right)|} \\
&+\frac{C}{|L\left(a\right)\cup L\left(b\right)||L\left(b\right)|}\sum_{\substack{a^{'}\in L(a)\setminus L(b)\\ b^{'}\in L(b)}} s\left(a^{'},b^{'}\right) \\
&++\frac{C}{|L\left(a\right)\cup L\left(b\right)||L\left(a\right)|}\sum_{\substack{b^{'}\in L(b)\setminus L(a)\\ a^{'}\in L(a)}} s\left(a^{'},b^{'}\right)
\end{aligned}
\tag{2.7}
$$

This has the advantage that there is no parameter analogous to λ in P-Rank that needs to be chosen beforehand, while still retaining the benefits of PSimRank.

2.5 SimRank*

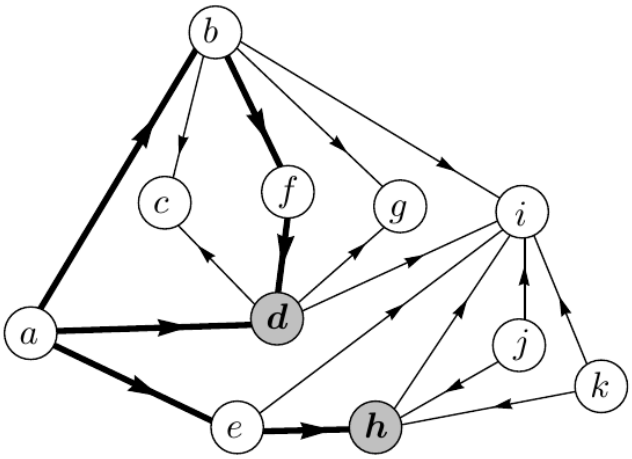

Figure 2.2: An example of the level-wise computation problem: There are no paths of equal length from a to d and h. This means that SimRank will ignore a as a source of similarity. Image source : Yu *et al.* (2013)

SimRank* (Yu *et al.* (2013)) was proposed to solve the third problem, the level-wise computation problem of SimRank. SimRank insists that a pair of nodes need to be at the same "level" from other nodes in order for the measure to be available for that pair. This means that it disregards any paths of unequal length meeting at a common node as evidence of similarity. An example of this is illustrated in Figure 2.2. SimRank* considers these paths with appropriate weights, leading to the following elegant matrix form :

$$\mathbf{S} = \frac{C}{2} \cdot \left(\mathbf{Q} \cdot \mathbf{S} + \mathbf{S} \cdot \mathbf{Q}^T\right) + (1 - C) \cdot \mathbf{I} \tag{2.8}$$

The recursive form as it turns out is even simpler than SimRank :

$$s\left(a, b\right) = \frac{C}{2|I(a)|} \sum_{i=1}^{|I(a)|} s\left(I_i(a), b\right) + \frac{C}{2|I(b)|} \sum_{i=1}^{|I(b)|} s\left(a, I_i\left(b\right)\right) \tag{2.9}$$

The measure is derived in Yu *et al.* (2013) by actually enumerating all pairs of paths of (possibly) unequal length from a and b to a common node and computing the weighted sum of an exponentially decayed score associated with each path. Later on in Section 3.4.5, we present a much simpler explanation as to how it works under the Random Surfer-Pair model interpretation.

Even using the naive iterations to compute SimRank* incurs only $O\left(N^3\right)$ complexity, whereas SimRank takes $O\left(N^4\right)$. A partial sums memoization approach is also proposed that reduces this further to only $O\left(N^2\right)$ in the worst case (which, it should be noted is the bare minimum if all pairs of similarities are to be computed). Thus, SimRank* is faster and at the same time resolves a major issue in SimRank.

2.6 MatchSim

MatchSim (Lin *et al.* (2012)) also solves the pairwise normalization problem, but in a different way. Instead of using the average similarity of *all* neighbors, only the most similar neighbor pairs are used. The most similar pairs are selected by finding the maximum matching between the neighbors of the two nodes, with the weight of the matching being the current estimate for the similarity. This leads to the following recursive form :

$$s\left(a,b\right)=\frac{\widehat{W}\left(a,b\right)}{\max\left(\left|I\left(a\right)\right|,\left|I\left(b\right)\right|\right)} \tag{2.10}$$

Where $\widehat{W}\left(a,b\right)$ is the total weight of the maximum matching between the neighbors of a and b.

2.7 CoSimRank

CoSimRank (Rothe and Schütze (2014)) has a matrix form not unlike the others discussed so far :

$$\mathbf{S}=C\cdot\left(\mathbf{Q}^T\cdot\mathbf{S}\cdot\mathbf{Q}\right)+\mathbf{I} \tag{2.11}$$

It turns out that this is equivalent to computing the following expressions:

$$s\left(a,b\right)=\sum_{k=0}^{\infty}c^k\left\langle p^{(k)}\left(a\right),p^{(k)}\left(b\right)\right\rangle \tag{2.12}$$

Where $p^{(k)}(a)$ is the Personalized PageRank (PPR) vector of node a computed using k iterations. This form can be computed element wise unlike all of the measures discussed so far, though it requires the PPR vectors to be available.

Chapter 3

GENERALIZED RANDOM SURFER-PAIR MODELS

Although SimRank is known by its recursive formulation, there is another useful, albeit less frequently discussed interpretation for SimRank known as the Random Surfer-Pair Model, which was proposed alongside SimRank (Jeh and Widom (2002)). PSimRank was another major work to use this interpretation. However, apart from these, it seems to have gone largely unnoticed. This probabilistic interpretation is the focus of the chapter and is discussed in detail in the upcoming section in the context of SimRank and PSimRank and later generalized to other measures.

3.1 Existing Random Surfer-Pair Models

3.1.1 SimRank

The Random Surfer-Pair interpretation for Simrank is based on a random experiment involving two random walks (or surfers) starting at the given nodes a and b, and traversing the graph *backwards* until they meet. That is, at the end of each step, each walk transitions to a randomly chosen in-neighbor. If either of the current nodes have no in-neighbors, the experiment is stopped.

Definition 1. Let $L(a, b)$ be the random variable that gives the number of steps taken until the surfers meet starting from a and b respectively. The *expected f-meeting distance* between a and b is defined for a given function f as $\mathbb{E}\left[f\left(L\left(a, b\right)\right)\right]$.

The f-meeting distance can be viewed as a score resulting from each instance of the experiment, and is itself a random variable. It turns out that for a specific choice of f, the expected score is nothing but the SimRank of (a, b). This equivalence of the Random Surfer-Pair formulation and the recursive form in equation 2.1 are stated in Jeh and Widom (2002) as the following theorem :

Theorem 2. *SimRank as defined by equation (2.1) is the same as the expected f-*meeting distance *between a and b for $f(t) = C^t$.*

If the experiment is stopped because of unavailability of neighbors, $L(a, b)$ is considered to be infinite, thus making the f-meeting distance zero for that run of the experiment. Also, the base case of a node being maximally similar to itself follows naturally because if $a = b$, the surfers deterministically meet at time $t = 0$, giving a score of 1 always.

In this interpretation, two nodes are similar if they are close to some source(s) of similarity.

3.1.2 PSimRank

In the Random Surfer-Pair Model, it becomes clearer as to how PSimRank solves the problem of pairwise normalization. It is done by increasing the tendency for the surfers to meet when they have more neighbors in common. Of course, as with SimRank, this model can be shown to be equivalent to its recursive form. The precise statement is as follows (Fogaras and Rácz (2005)) :

Theorem 3. *The recursive PSimRank score between nodes a and b defined by equation (2.6) is nothing but the expected f-meeting distance with $f(t) = C^t$ of two random surfers X_a and X_b starting from a and b that move in the following way. If $X_a(t) = u$ and $X_b(t) = v$ are their positions at some time t :*

- *With probability $\frac{|I(u) \cap I(v)|}{|I(u) \cup I(v)|}$ (which is the Jaccard coefficient of the sets of in-neighbors of u and v), they both move to the same uniformly chosen node in the set of common in-neighbors $I(u) \cap I(v)$.*
- *With probability $\frac{|I(u) \setminus I(v)|}{|I(u) \cup I(v)|}$, X_a moves to a uniformly chosen node in $I(u) \setminus I(v)$ and the walk X_b steps to an independently chosen uniform vertex in $I(v)$.*
- *With probability $\frac{|I(v) \setminus I(u)|}{|I(u) \cup I(v)|}$, X_a moves to a uniformly chosen node in $I(u)$ and the walk X_b steps to an independently chosen uniform vertex in $I(v) \setminus I(u)$.*

One thing to note here is that the surfers are now coupled as opposed to being independent in the case of SimRank. That is, the transitions of one surfer do influence the other.

3.2 Generalizing the Random Surfer-Pair Model

An interesting observation to be made is that PSimRank was formulated as a Random Surfer-Pair Model and then cast as a recursive definition, in contrast to SimRank. Given its success, it is natural to consider the possibility of a general version that can be applied to many of the measures discussed here.

The generalization that will be presented here treats the Random Surfer-Pair experiment as a single random walk, but on a compound state space that consists of vertex pairs from $V \times V$ to indicate the positions of both surfers, and also a "stopped" state, which represents unavailability. We use the letter h to denote a typical state from this space, which we denote by $\mathcal{S}$.

The transition probabilities for this random walk are denoted as $p\left(h^{'} \mid h\right)$, the probability of transitioning to $h^{'}$ from h. The stopped state is an absorbing state, that is once the state is reached, it is impossible to leave. We can collect these probabilities into a matrix $\mathbf{P}$.

The idea is that different measures can be realized for different choices of transition probabilities, formalized by the following definition of the Generalized Random Surfer-Pair Model :

Definition 4. For a particular matrix of transition probabilities $\mathbf{P}$, consider the following combined random walk experiment over the compound state space $\mathcal{S}$ starting from (a, b) at time $t = 0$:

- If the current state of the walk is h, the walk moves to the next state with probability $p\left(h^{'} \mid h\right)\ \forall\, h^{'} \in \mathcal{S}$ as specified by $\mathbf{P}$.
- The walk ends when a state of the form (x, x) is reached, for some $x \in V$.

Let $L(a, b)$ be the random variable that gives the number of steps taken until the walk ends. The *expected f-meeting distance* between a and b is defined for this combined walk for the function $f(t) = C^t$ as $\mathbb{E}\left[f\left(L\left(a, b\right)\right)\right]$. This is a function of (a, b) which we call the *similarity measure induced by* $\mathbf{P}$ under the Generalized Random Surfer-Pair model.

The termination condition is equivalent to the random surfers meeting at some node for the first time. If the walk goes into the stopped state, it stays there forever, and does not reach a state of the form (x, x). This gives an infinite number of steps, thus leading to a score of zero. Again, the base case of maximal self-similarity applies here as well.

3.3 Equivalence to recursive form

It is straightforward to see that the coefficients of any $s\left(a^{'}, b^{'}\right)$ in the recursive formulations of SimRank (Equation 2.1) and PSimRank (Equation 2.6) are the same as the transition probabilities for the surfers in their respective Random Surfer-Pair Models going from (a, b) to $\left(a^{'}, b^{'}\right)$. This leads one to believe there could be a similar relationship to a recursive form for any transition probabilities $\mathbf{P}$. Indeed, this is true and the results are formally presented in the remainder of this section.

For a given transition probability matrix $\mathbf{P}$, consider the following set of recursive equations defined for all node pairs (a, b) :

$$s\left(a, b\right) = C \sum_{\left(a^{'}, b^{'}\right) \in \mathcal{R}((a,b))} p\left(\left(a^{'}, b^{'}\right) \,\middle|\, (a, b)\right) s\left(a^{'}, b^{'}\right) \tag{3.1}$$

Here, $\mathcal{R}\left((a, b)\right)$ is a region of support (which we will also refer to as *support set*) for (a, b) under $\mathbf{P}$, that is where the transition probability $p\left(\left(a^{'}, b^{'}\right) \,\middle|\, (a, b)\right)$ is non-zero. Note that this *does not* include the stopped state, which means the sum of the coefficients appearing in the above equation need not be 1 (of course, they have to be less than 1).

The same base case of $s\left(a, a\right) = 1 \,\forall\, a \in V$ is used. If $\mathcal{R}\left((a, b)\right) = \{\phi\}$, $s\left(a, b\right)$ is taken to be zero unless $a = b$. These equations define what is called the *recursive similarity measure induced by* $\mathbf{P}$.

As before, an iterative form is also defined :

$$s_{k+1}(a,b) = C \sum_{\left(a',b'\right)\in\mathcal{R}((a,b))} p\left(\left(a',b'\right)\Big|(a,b)\right) s_k\left(a',b'\right) \tag{3.2}$$

Theorem 5. *The following results hold true for this iterative form :*

- ***Monotonicity and boundedness** :*

$$0 \leq s_k(a,b) \leq s_{k+1}(a,b) \leq 1 \qquad \forall\ (a,b) \in V \times V$$

- ***Convergence to limit** : The sequence $s_k(a,b)$ converges to a limit (obviously between 0 and 1 by the previous part) for all $(a,b) \in V \times V$*

From the above, the following result follows :

Theorem 6. *There exists a unique solution to the system of equations defined by equation 3.1.*

Which leads to the following central result :

Theorem 7. *The similarity measure induced by $\mathbf{P}$ according to definition 4 is the same as the recursive similarity measure induced by $\mathbf{P}$ (Equation 3.1).*

The above theorems generalize the results in Jeh and Widom (2002). The proofs are also done in a similar manner and are presented in Appendix A.

Theorem 7 is the result necessary to convert an existing recursive form into a Random Surfer-Pair Model. All that needs to be done is to read off the non-zero coefficients into the appropriate places into the matrix, or equivalently get the support set and the corresponding transition probabilities as a function of (a,b). One thing to note here is that the probabilities corresponding to *actual* node pair destinations from (a,b) need not sum to 1, because it could go into the stopped state as well.

It would of course be more illustrative to get a concise description of the matrix. Our formulation allows for transitions from one compound state to any arbitrary state, and any number of destination states with non-zero transition

probability. However, as we will see in the upcoming sections, in existing measures, there are only a few possible transitions from any given state (a, b), and that too involving the neighbor pairs of a and b. This means that $\mathbf{P}$ is usually sparse, so even though there are N^2 states, only a few of them are involved in transitions from any given state, and it is no more complicated than the existing recursive formulations. However, we note that the above results continue to hold for any $\mathbf{P}$ regardless of sparsity.

3.4 Reinterpreting Existing Measures

3.4.1 SimRank

- Support set : $\mathcal{R}((u, v)) = I(u) \times I(v)$.
- Transition probabilities : $\frac{1}{|I(u) \times I(v)|} \quad \forall\, x \in I(u) \times I(v)$

$\mathcal{R}((a, b))$ is simply all neighbor pairs of u and v with the transition probabilities being uniform over this set. If either node has no in-neighbors, it transitions to the stopped state with probability 1 (i.e unavailable).

3.4.2 PSimRank

From the Random Surfer-Pair formulation in Theorem 3, it is straightforward to find :

- Support set : $\mathcal{R}((u, v)) = I(u) \times I(v)$.
- Transition probabilities :
  - For common in-neighbors :

$$p((x, x) \,|\, (u, v)) = \frac{1}{|I(u) \cup I(v)|} \quad \forall\, x \in I(u) \cap I(v)$$

  - For first surfer not choosing common neighbor :

$$p\left(\left(u', v'\right) \,\middle|\, (u, v)\right) = \frac{1}{|I(u) \cup I(v)||I(v)|} \quad \forall\, u' \in I(u) \setminus I(v),\ \forall v' \in I(v)$$

  - For second surfer not choosing common neighbor :

$$p\left(\left(u', v'\right) \,\middle|\, (u, v)\right) = \frac{1}{|I(u) \cup I(v)||I(u)|} \quad \forall\, v' \in I(v) \setminus I(u),\ \forall u' \in I(u)$$

Again, if either of the nodes do not have in-neighbors, the walk goes to the stopped state like SimRank.

3.4.3 C-Rank

C-Rank being an undirected variant of PSimRank, the Random Surfer-Pair Model follows right away from PSimRank by replacing in-neighbors with the set of undirected neighbors everywhere.

3.4.4 P-Rank

From definition 2.5, we have :

- Support set : $\mathcal{R}((a,b)) = (I(a) \times I(b)) \cup (O(a) \times O(b))$
- Transition probabilities :
  - In-neighbors : $p\left(\left(a^{'},b^{'}\right) \middle| (a,b)\right) = \frac{\lambda}{|I(a)\times I(b)|} \quad \forall \left(a^{'},b^{'}\right) \in I(a) \times I(b)$. If $|I(a)| = 0$ or $|I(b)| = 0$, it goes to the stopped state instead with probability λ.
  - Out-neighbors : $p\left(\left(a^{'},b^{'}\right) \middle| (a,b)\right) = \frac{1-\lambda}{|O(a)\times O(b)|} \quad \forall \left(a^{'},b^{'}\right) \in O(a) \times O(b)$. If $|O(a)| = 0$ or $|O(b)| = 0$, it goes to the stopped state instead with probability $1-\lambda$.

This is essentially the following : a coin with probability λ is tossed, and based on its result, *both* surfers move backward or forward and choose from applicable edges uniformly.

3.4.5 SimRank*

From definition 2.9, we have :

- Support set : $\mathcal{R}((a,b)) = (\{a\} \times I(b)) \cup (I(a) \times \{b\})$
- Transition probabilities :

$$p\left(\left(a^{'},b^{'}\right) \middle| (a,b)\right) = \begin{cases} \frac{1}{2|I(a)|} & \forall \left(a^{'},b^{'}\right) \in \{a\} \times I(b) \\ \frac{1}{2|I(b)|} & \forall \left(a^{'},b^{'}\right) \in I(a) \times \{b\} \end{cases}$$

The unavailability of in-neighbors is treated the same way as SimRank, except that one of the two cases above can still be "active" and not go to the stopped state in situations where $|I(a)| = 0,\ |I(b)| \neq 0$ or vice-versa.

The notable feature here is that only one of the surfers is allowed to move at each step. The choice as to which surfer moves is made uniformly. This can be summed up as : toss a fair coin, based on the outcome, one surfer chooses an in-neighbor uniformly.

From this, it becomes clear how SimRank* manages to consider paths of unequal length. The surfers need not have made an equal number of jumps to meet at some node. By allowing only one to move at a time, it is ensured that all pairs of paths from (a, b) to a common node that have the same total length also have the same weight or probability.

3.5 Monte Carlo Computation

The Monte Carlo paradigm is a widely used one across many fields because it is applicable whenever a quantity can be computed as an expected value. Naturally, it also applies to the Generalized Random Surfer-Pair model. A straightforward way to use it would be to simulate the random walk starting from a given pair of nodes for some number of times, and return the average score as the similarity.

In practice, the surfers would have to be truncated after some of steps L_{max}, and only a limited number of samples N_S can be drawn in the interest of fast querying, but Fogaras and Rácz (2005) provide decent guarantees for accuracy in practice. As with all the recursive measures, radius based pruning can be done to reduce the amount of nodes that need to be considered for top-k similarity queries.

Typically, there is an easy way to generate a transition from any given state in constant time. In SimRank* for instance, all that needs to be done is to toss a coin, and advance one surfer to a randomly chosen in-neighbor. Therefore, the complexity of a single similarity computation is just $O(N_S L_{max})$, where these quantities are much smaller than the size of the network.

Even with more complicated transitions like those involving the Jaccard Coefficient in PSimRank, it is possible to use efficient indexing for fast querying. The indexing approach presented in Fogaras and Rácz (2005) is based on generating several coalescing random walks and organizing them in a compact manner in what is called a Fingerprint Graph. This structure enables fast querying of first meeting times starting from any two nodes. However, this approach might not work for all kinds of transitions. For instance, with SimRank*, since at any step, only one of the surfers can move, it would be impossible to generate a set of coalescing random walks in a consistent manner because in a typical set of such walks, there would be many pairs where both move or neither move at some step. Nevertheless, this is a highly useful approach, specially in the setting of an actual query engine.

3.6 P+Rank : A more consistent measure

The Random Surfer-Pair formulation for P-Rank presented in section 3.4.4 is essentially the following : a coin with probability λ is tossed, and based on its result, *both* surfers move backward or forward and choose from applicable edges uniformly. This coupling of the walks seems to be unnecessary, and in theory discards common sources of similarity that would be reachable if each surfer could choose direction independently.

The straightforward way to remove the coupling would be to make the surfers choose a direction independently, using two separate coin tosses. Thus, there is a possibility for one surfer to move forward and the other backward, or vice-versa. This measure is termed P+Rank. The equivalent recursive formulation for P+Rank would be :

$$
\begin{aligned}
s\left(a,b\right) = \lambda^2 \times \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s\left(I_i(a), I_j(b)\right) \\
+ \lambda(1-\lambda) \times \frac{C}{|I(a)||O(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|O(b)|} s\left(I_i(a), O_j(b)\right) \\
+ (1-\lambda)\lambda \times \frac{C}{|O(a)||I(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|I(b)|} s\left(O_i(a), I_j(b)\right) \\
+ (1-\lambda)^2 \times \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} s\left(O_i(a), O_j(b)\right) \quad (3.3)
\end{aligned}
$$

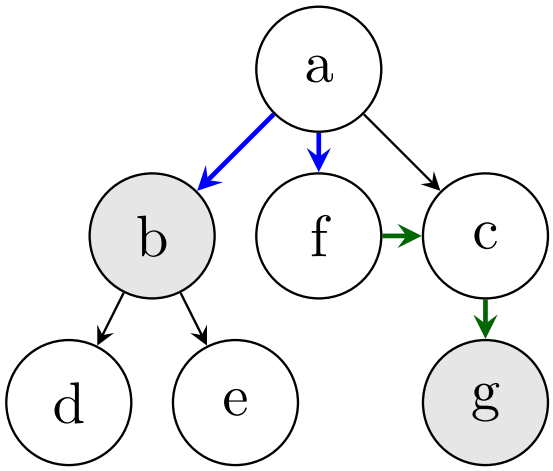

Figure 3.1: Example of P-Rank ignoring paths because only one of them changes direction : P-Rank is unavailable for (b, g) because there are no paths that follow the constraint that both must choose the same direction. P+Rank however considers the paths indicated in blue and green.

This change makes P+Rank consider more common sources of similarity that P-Rank doesn't. An example of this is a situation where the path of *exactly one* of the surfers needs to change direction in order to meet at a source of similarity, illustrated in Figure 3.1. Thus, P+Rank is theoretically more consistent as a direction aware measure (and direction awareness is the improvement that P-Rank makes over SimRank).

P+Rank also has the advantage that because of the surfers being independent,

the Monte Carlo indexing algorithm in Fogaras and Rácz (2005) can be used directly. All that needs to be done to adapt the algorithm for P+Rank is to generate the fingerprints by choosing edges as described above. This will not work for P-Rank, where the coupling of the surfers means that the random fingerprints used in the indexing can't be consistent. That is, there will be coalescing random walks that move in different directions, which is not allowed in P-Rank.

Although Equation 3.3 appears to be more complicated than the recursive form for P-Rank in Equation 2.5, both measures take nearly the same time when using Monte Carlo methods, with P+Rank requiring only an additional coin toss at each step.

3.7 PSimRank* : Combining the Best of Both Worlds

Previously, we have described how PSimRank solves the Pairwise Normalization problem, and SimRank* solves the Level Wise Computation problem, and what they entail in the Random Surfer "domain", so to speak. Now, we attempt to combine these two benefits in the hope that it will result in a better measure because of solving both the problems (the name is a portmanteau of PSimRank and SimRank*).

The combination is straightforward; to make the surfers meet at a common in-neighbor with probability equal to the Jaccard coefficient just like in PSimRank, but the remainder of the time behave like SimRank*, moving only one at a time. Therefore, the Random Surfer-Pair model for this is given by :

- Support set : $\mathcal{R}\left((a,b)\right) = I\left(a\right) \times I\left(b\right)$.
- Transition probabilities :
  - For common in-neighbors :

$$p\left((x,x)\,|\,(a,b)\right) = \frac{1}{|I\left(a\right) \cup I\left(b\right)|} \quad \forall\, x \in I\left(a\right) \cap I\left(b\right)$$

  - If not choosing a common neighbor, behave like SimRank* :

$$p\left(\left(a^{'},b^{'}\right)\,\middle|\,(a,b)\right) = \begin{cases} \left(1 - \frac{|I(a)\cap I(b)|}{|I(a)\cup I(b)|}\right) \times \frac{1}{2|I(a)|} & \forall\, \left(a^{'},b^{'}\right) \in \{a\} \times I\left(b\right) \\ \left(1 - \frac{|I(a)\cap I(b)|}{|I(a)\cup I(b)|}\right) \frac{1}{2|I(b)|} & \forall\, \left(a^{'},b^{'}\right) \in I\left(a\right) \times \{b\} \end{cases}$$

The equivalent recursive form for this would be :

$$
\begin{aligned}
s\left(a,b\right) &= \frac{C|I\left(a\right)\cap I\left(b\right)|}{|I\left(a\right)\cup I\left(b\right)|} \\
&+ \left(1-\frac{|I\left(a\right)\cap I\left(b\right)|}{|I\left(a\right)\cup I\left(b\right)|}\right)\left(\frac{C}{2|I(a)|}\sum_{i=1}^{|I(a)|}s\left(I_i(a),b\right)+\frac{C}{2|I(b)|}\sum_{i=1}^{|I(b)|}s\left(a,I_i\left(b\right)\right)\right)
\end{aligned}
\tag{3.4}
$$

Where we have used $s\left(x,x\right)=1$ in the first term.

3.8 Experiments

In this section, the two new measures that have been discussed, P+Rank and PSimRank* are put to test on a real-world dataset alongside many of the existing measures. P+Rank and P-Rank being parametric, a sweep over the λ parameter is performed for both and the best performing value of λ is used for comparison. The other measures are all non-parametric. Reported for comparison are the scores for SimRank, Rvs-SimRank, PSimRank and SimRank*.

We use the Arnetminer dataset (Tang *et al.* (2008)), which is a citation network of papers extracted from DBLP[1]. A portion of these papers have been manually annotated and given labels corresponding to 10 different topics (clusters). The evaluation consists of running a top-k similarity query on some randomly chosen labeled nodes, and finding the Mean Average Precision (MAP) (Manning *et al.* (2008)) for the answer set having the same label as the query. The rationale behind this is that papers in the same topic as the query are likely to be similar.

For all measures used here, pruning was done to radius 4 (in the undirected graph). The Random Surfer simulations were performed 200 times per query, and truncated after at most 15 steps. Top-100 queries were run on 50 randomly chosen labeled nodes that had at least 5 citations and 5 references to ensure that

[1]DBLP website

λ	MAP: P-Rank	MAP: P+Rank
0.1	0.75	0.76
0.2	0.75	0.76
0.3	0.76	0.75
0.4	0.76	0.71
0.5	0.76	0.73
0.6	0.72	0.73
0.7	0.73	0.72
0.8	0.71	0.72
0.9	0.70	0.73

Measure	Best MAP
SimRank	0.73
Rvs-SimRank	0.71
P-Rank	0.76
P+Rank	0.76
SimRank*	0.80
PSimRank	0.80
PSimRank*	0.81

Table 3.1: Left : MAP values attained for various values of λ by P-Rank and P+Rank. Right : Best MAP values compared to other measures.

the measures wouldn't become unavailable. Since not all the nodes are labeled, only the nodes in the answer set that have a label are considered for calculating the MAP scores. Further, 50 such trials are performed and the averaged MAP scores are reported in Table 3.1.

It is observed that PSimRank* outperforms all the other measures, improving on both PSimRank and SimRank* on which it is based, thus demonstrating the effectiveness of combining the behavior of PSimRank and SimRank* in the Random Surfer-Pair interpretation.

Additionally, PSimRank and SimRank* outperform all others except PSimRank*, indicating that solving the pairwise normalization problem and the level-wise computation problem brings a greater benefit (more so when both are solved together as in PSimRank*) than just taking into account directions like in P-Rank and P+Rank.

Regarding the comparison between P-Rank and P+Rank, we see that they are both tied in their best MAP scores, with 0.76. Thus, P+Rank performs at least as well as P-Rank while being a more theoretically sound measure. P-Rank and P+Rank both perform better than SimRank and Rvs-SimRank (in their best scores after parameter tuning), which shows that there is some benefit to considering out-links, just not as much as solving the other two issues.

It is worthwhile to note that these experiments could be run on a commod-

ity PC with a 4 core Intel i5 processor, with each query finishing in about a second. That is, about 100,000 similarity computations (after pruning) could be performed in that time frame. On a heavy duty machine with more CPUs and more cores per CPU, this number only increases, thus demonstrating the computational efficiency and scalability of the Monte Carlo approach. The implementations were made in C++ with the Boost Graph Libraries running on Ubuntu 14.04 LTS.

3.9 Other Possibilities :

3.9.1 Incorporating edge weights :

The model proposed allows for any distribution over the next states. These probabilities could be obtained from normalized edge weights, or using a softmax function. Thus, it would for example be possible to incorporate a text based similarity measure through edge weights.

3.9.2 Termination and Scoring :

Scope for further generalization exists in the scoring and termination criteria used in the experiment. For instance, if one were to allow the walks to continue indefinitely regardless of how many times they meet, but change the scoring to add $f(t)$ to the score for each time t at which the surfers meet, this would emulate a measure like Generalized Co-Citation (Narwekar (2016)) but with probabilistic weights for the paths. Also a version that considers paths of unequal lengths can also be made by applying this in conjunction with the "one-at-a-time" principle of SimRank*.

3.10 Benefits of the Generalized Random Surfer-Pair Model

3.10.1 Unification and Generalization

Several seemingly disparate measures have been brought under the same unifying framework. Any properties that are discovered for this model would also apply to these measures.

The model has helped create a hybrid measure, PSimRank* that successfully combines the benefits of two different measures, PSimRank and SimRank* that have different properties to create a better performing measure.

The framework can also be used in many other ways, as illustrated in section 3.9, opening up possibilities for discovering many new classes of measures.

3.10.2 New Insights Into Existing Work

By reinterpreting the measures under this framework, salient features are brought to light that aren't at all obvious in the recursive formulations :

- A quirk of P-Rank is exposed wherein both surfers have to move in the same direction. This is not at all evident from the recursive form in Equation 2.5.
- The original derivation for SimRank* (Yu *et al.* (2013)) that analytically computes a measure that includes paths of unequal lengths is involved and laborious. Under this interpretation however, it becomes intuitive as to why it works.

3.10.3 Computational Advantages

By bringing several existing measures under this model, Monte Carlo methods can be applied to compute them. Therefore, the advantages of Monte Carlo methods are also inherited. We present these advantages below.

The first advantage that comes to mind is intrinsic to the Monte Carlo framework, and that is extensive parallelizability of simulations : every instance of the random experiment can be run separately and concurrently.

Typically in the context of similarity measures, one important application is a top-k query, which would involve computation of similarity between many pairs of nodes. Under Random Surfer-Pair models, these computations can all be performed independently, unlike when solving the entire system of recursive similarity equations. This adds yet another layer of parallelizability.

Next is the dramatically low memory requirement : only a constant amount of memory is needed for each similarity being computed. All pairs of node similarities are almost never required to be computed, and only much fewer than the total $\binom{N}{2}$ similarities are required. Solving the recursive forms on the other hand would mean having to store all of them, and the $O(N^2)$ memory requirement would be prohibitive even for medium sized graphs with $N \approx 10^6$.

Monte Carlo methods also have an advantage in terms of computation time: it is independent of the size of the network.

3.11 Conclusions :

A generalized Random Surfer-Pair model has been developed which subsumes many well known similarity measures in one unifying framework. Admittedly, it is not all-encompassing; it is not at all evident how it can be applied to MatchSim and CoSimRank. Reinterpreting P-Rank and SimRank* under this framework has provided interesting insights as to their functioning.

Working under this interpretation has enabled the development of a better performing measure, namely PSimRank*. A theoretical deficiency in P-Rank was also exposed and easily remedied under this framework. Most importantly, it has enabled Monte Carlo methods to be applied to these measures, along with many computational advantages. Hopefully, this work has opened up many

possibilities for theoretical dissection and development of better measures.

Chapter 4

NEGATIVE EXAMPLES IN RECOMMENDATIONS

Academic search engines such as Google Scholar, Microsoft Academic Search and CiteSeerX are becoming increasingly important as tools for performing research efficiently. Providing recommendations based on query papers would obviously be a fundamental part of these systems. These recommendations can be made more effective if it were possible to specify not just papers that the user considers relevant, but also papers that are not like what the user requires, i.e negative examples. This problem is formally stated as follows for recommenders that use some notion of similarity :

Recommendations with negative examples : Given a database of items $\mathcal{D}$, a seed set of positive examples $X \subseteq \mathcal{D}$, and a negative seed set $Y \subseteq \mathcal{D}$, recommend a set of items $\mathcal{R}$ such that each item $r \in \mathcal{R}$ is similar to as many items in X as possible while being dissimilar to as many items in Y as possible.

An added feature would be to be able to set user preferences for each of the seed examples. Most importantly, for such capabilities to be most useful, the method would have to be flexible in allowing how many seeds of each type are allowed. That is, it should be possible for example to give only negative examples with no positive examples.

4.1 Possibilities for handling negative examples with link based measures

In this section, we investigate possible ways of adapting structural similarity measures to enable them to make use of negative examples.

There appears to be very little existing literature along these lines to the best of the author's knowledge. To the best of the author's knowledge, only one work (Küçüktunç *et al.* (2012)) even attempts to use this manner of relevance feedback. The way it deals with it is also rather trivial; it suggests simply dropping nodes from the graph when marked as a negative example and then issuing queries on the changed graph. Still, this could be useful as a baseline for future work.

Absorbing random walks (Mavroforakis *et al.* (2015); Singh *et al.* (2007)) seem at first to be a good prospect to use in a Personalized PageRank setup : the negative examples in Y could be made into absorbing states for the PPR random walk, while X is the set of starting nodes, with a ranking based on the expected number of visits by the random walk before getting absorbed. The fatal flaw of such an approach is that it is ill defined when there are no X or Y nodes, that is it does not have the flexibility described earlier.

A straightforward way to use any structural similarity measure is to use a weighted sum of similarities of a query node q with positive weights for X nodes and negative weights for Y nodes as a score like so :

$$\sum_{x \in X} w_x s(x, q) - \sum_{y \in Y} w_y s(y, q) \tag{4.1}$$

Where w_x and w_y are normalizing weights that could be based on user preference.

4.2 PacRank

Equation 4.1 involves using a similarities that are computed for each seed node completely independently of the other seed nodes. We propose PacRank[1], an algorithm where nodes from X and Y *compete* for sources of similarity with q

[1]The name is an allusion to the famous video game Pacman, with the random surfers experiment being reminiscent of the ghosts chasing the player in the game.

rather than working separately. It is an extension of the Random Surfer-Pair model, with the following changes :

- There are now *three* random surfers :
  - One starting at a node in X with probabilities based on given preferences, which we will refer to as the X *surfer*.
  - One starting at a node in Y with probabilities based on given preferences, which we will refer to as the Y *surfer*.
  - A third walk starting at query node q (*query surfer*).
  - If there are no X or Y nodes, the corresponding surfer is simply dispensed with.
- The experiment ends when the query surfer is "caught" by either the X or the Y surfer, after L steps.
- The score is the expected value of a random variable S defined as follows :
  - $S = C^L$if caught by the X walk.
  - $S = -C^L$if caught by the Y walk.

In this "Random Surfer-Triplet" model, the X and Y surfers try to "capture" sources of similarity, and the score helps to discriminate between them in terms of how often they can do so.

Unfortunately, there are hurdles preventing effective evaluation of any method for handling negative examples. First is the lack of any publicly available annotated dataset that is large enough. User studies to evaluate the quality of recommendations are also expensive and time-consuming to conduct. Unlike simple structural similarity measures, there is also no straightforward "intrinsic" way to evaluate effectiveness like the cluster based evaluation that was performed earlier. Thus, while we theorize that PacRank should perform better, we do not perform evaluation in this work.

4.3 Incremental Feedback Based Querying :

A search engine interface would allow the user to give relevance feedback dynamically, that is to mark part of the results as positive or negative examples, and then receive updated results. This would require the algorithm to be able

to incrementally update results.

In PacRank, we exploit the fact that the score for given seed sets X and Y is simply a weighted sum of scores w.r.t each pair $(x, y) \in X \times Y$. That is, run PacRank assuming $X = \{x\}$ and $Y = \{y\}$ for possible pairs $(x, y) \in X \times Y$, and then take the weighted average. The weights are nothing but the probabilities of the X and Y surfers starting from x and y respectively. This follows directly from the linearity of expectation used on the first step.

All that needs to be done is to store such scores for each query node, for each of the pairs $(x, y) \in X \times Y$. Note that not all nodes in the network need to be involved when pruning is done, and only scores for the nodes that are not pruned need be stored. Typically, since the seed sets are specified by the user, their size would be in comparison to the entire network $O(1)$, so the memory requirements would be reasonable throughout the search session.

When feedback is given for a new node, scores are computed for the pairs involving that node, and the full scores are updated incrementally with these new scores.

4.4 Conclusions

There has been little work done on handling negative examples with structural similarity measures. One way would be to use a (signed) weighted combination of similarities as scores to induce a ranking. An approach that discriminates between candidate nodes based on how much they tend to capture sources of similarity is proposed, namely PacRank. An incremental version of this algorithm is also given for applying in a real time querying setting. Although this is in theory a viable approach, evaluation could not be performed due to several problems.

Appendix A

PROOFS OF THEOREMS

A.1 Theorem 5

The monotonicity and boundedness are proved by induction. The inductive hypothesis is that

$$0 \leq s_{k-1}(a,b) \leq s_k(a,b) \leq 1 \qquad \forall\ (a,b) \in V \times V$$

The base case of this for $k = 0$ is trivial since $s_0(x,x) = 1$ and $s_0(a,b) = 0$ $\forall\, a \neq b$, and so is the case with $a = b$. The inductive step is as follows :

A.1.1 Monotonicity :

We have

$$s_{k+1}(a,b) - s_k(a,b) = C \sum_{\left(a^{'},b^{'}\right) \in \mathcal{R}((a,b))} p\left(\left(a^{'},b^{'}\right) \Big| (a,b)\right)\left[s_k\left(a^{'},b^{'}\right) - s_{k-1}\left(a^{'},b^{'}\right)\right]$$

But $s_k\left(a^{'},b^{'}\right) - s_{k-1}\left(a^{'},b^{'}\right) \geq 0$ by the inductive hypothesis, and $p\left(\left(a^{'},b^{'}\right) \Big| (a,b)\right) \geq 0$ since it is a probability, thus proving the monotonicity.

A.1.2 Boundedness :

From the inductive hypothesis, $0 \leq s_k(a,b) \leq 1$. Therefore,

$$\begin{aligned} s_{k+1}(a,b) &= C \sum_{\left(a^{'},b^{'}\right) \in \mathcal{R}((a,b))} p\left(\left(a^{'},b^{'}\right) \Big| (a,b)\right) s_k\left(a^{'},b^{'}\right) \\ &\leq C \sum_{\left(a^{'},b^{'}\right) \in \mathcal{R}((a,b))} p\left(\left(a^{'},b^{'}\right) \Big| (a,b)\right) \cdot 1 \\ &\leq C \leq 1 \end{aligned}$$

Where we have used the fact that $\sum_{\left(a^{'},b^{'}\right)\in\mathcal{R}((a,b))} p\left(\left(a^{'},b^{'}\right)\Big|(a,b)\right)\leq 1$, since it is a sum of transition probabilities out of (a,b) (possibly less than one because of the stopped state). Similarly, it can be shown that $s_{k+1}(a,b)\geq 0$.

A.1.3 Convergence :

Since $s_k(a,b)$ is bounded and non-decreasing, by the Completeness Axiom of Calculus, $s_k(a,b)$ converges to a limit $\forall\ (a,b)\in V\times V$, which we denote by $g(a,b)$. Of course, this limit must be between 0 and 1 as the sequence itself is bounded in that range.

A.2 Theorem 6

First, we show that a solution exists for the recursive form and then that there cannot be two different solutions.

A.2.1 Existence

From the above, we know that $s_k(a,b)$ converges to a limit $g(a,b)$. So passing to the limit:

$$\lim_{k\to\infty} s_{k+1}(a,b) = C\sum_{\left(a^{'},b^{'}\right)\in\mathcal{R}((a,b))} p\left(\left(a^{'},b^{'}\right)\Big|(a,b)\right)\lim_{k\to\infty} s_k\left(a^{'},b^{'}\right)$$

Since $\lim_{k\to\infty} s_{k+1}(a,b) = \lim_{k\to\infty} s_k(a,b) = g(a,b)$, we have

$$g(a,b) = C\sum_{\left(a^{'},b^{'}\right)\in\mathcal{R}((a,b))} p\left(\left(a^{'},b^{'}\right)\Big|(a,b)\right) g\left(a^{'},b^{'}\right)$$

Thus, $g(a,b)$ is a solution to the general recursive form of Equation 3.1.

A.2.2 Uniqueness

Let two solutions to Equation 3.1 be s_1 and s_2. Define their difference

$$\delta(a,b) = s_1(a,b) - s_2(a,b)$$

Let M be the maximum absolute value of δ, that is $\max_{(a,b)} |\delta(a,b)|$. Let this maximum value be achieved for (a,b), that is $M = |\delta(a,b)|$. If $a = b$, then clearly $M = 0$ as both s_1 and s_2 must satisfy the maximal self-similarity base case. Otherwise, we have

$$\begin{aligned}
M &= \left| C \sum_{(a',b') \in \mathcal{R}((a,b))} p\left(\left(a',b'\right) \,\middle|\, (a,b)\right) \left[s_1\left(a',b'\right) - s_2\left(a',b'\right) \right] \right| \\
&= \left| C \sum_{(a',b') \in \mathcal{R}((a,b))} p\left(\left(a',b'\right) \,\middle|\, (a,b)\right) \delta\left(a',b'\right) \right| \\
&\leq C \sum_{(a',b') \in \mathcal{R}((a,b))} p\left(\left(a',b'\right) \,\middle|\, (a,b)\right) \left|\delta\left(a',b'\right)\right| \\
&\leq C \sum_{(a',b') \in \mathcal{R}((a,b))} p\left(\left(a',b'\right) \,\middle|\, (a,b)\right) M \\
&\leq CM
\end{aligned}$$

Here, we have used the fact that since (a,b) maximizes $|\delta(\cdot,\cdot)|$, $\left|\delta\left(a',b'\right)\right| \leq M$ and again the fact that $\sum_{(a',b') \in \mathcal{R}((a,b))} p\left(\left(a',b'\right) \,\middle|\, (a,b)\right) \leq 1$.

Now, since M is an absolute value, and $M \leq CM$ with $C < 1$, we must have $M = 0$. This proves that $s_1$and s_2 are always the same. Thus, there exists a unique solution to the recursive form of Equation 3.1, and that solution can be obtained as the limit of the iterative form.

A.3 Theorem 7

We first need to show that the expected f-meeting distances, for which we overload the notation $s\left(a,b\right)$ satisfy the recursive form. Let $\mathcal{W}_{a,b}$ be the set of all compound walks from (a,b) to a state of the form (x,x). Let $l\left(w\right)$ denote the length of such a walk w, and $p\left(w\right)$ the total probability, which is the product of the probabilities of the individual transitions. Then, by definition of the expected f-meeting distance,

$$s\left(a,b\right)=\sum_{w\in\mathcal{W}_{a,b}}p\left(w\right)C^{l(w)} \tag{A.1}$$

Now, consider the set of all such compound walks from one step ahead, that is, $\bigcup_{\left(a',b'\right)\in\mathcal{R}((a,b))}\mathcal{W}_{a',b'}$. Note that all the individual sets $\mathcal{W}_{a',b'}$ are disjoint. Now, clearly this collection of walks differs from $\mathcal{W}_{a,b}$ only in the inclusion of the first transition to some $\left(a^{'},b^{'}\right)$. Therefore, a bijection exists between this set and $\mathcal{W}_{a,b}$, and so $\mathcal{W}_{a,b}$ can be enumerated in terms of the new collection.

This means that for every member w of $\mathcal{W}_{a,b}$, there is some unique $\left(a^{'},b^{'}\right)$ and some unique $w^{'}\in\mathcal{W}_{a',b'}$. Thus, it is possible to group the terms of the summation in Equation A.1 by $\left(a^{'},b^{'}\right)$. Now, the corresponding $w^{'}$ will have one step lesser, so $l\left(w^{'}\right)+1=l\left(w\right)$, and it omits the transition probability for the step from (a,b) to $\left(a^{'},b^{'}\right)$, so

$$p\left(w\right)=p\left(\left(a^{'},b^{'}\right)\middle|\left(a,b\right)\right)p\left(w^{'}\right)$$

Thus, Equation A.1 is rewritten as:

$$\begin{aligned}s\left(a,b\right) &= \sum_{\left(a',b'\right)\in\mathcal{R}((a,b))}\sum_{w'\in\mathcal{W}_{a',b'}}p\left(\left(a^{'},b^{'}\right)\middle|\left(a,b\right)\right)p\left(w^{'}\right)C^{l\left(w^{'}\right)+1}\\ &= C\sum_{\left(a',b'\right)\in\mathcal{R}((a,b))}p\left(\left(a^{'},b^{'}\right)\middle|\left(a,b\right)\right)\sum_{w'\in\mathcal{W}_{a',b'}}p\left(w^{'}\right)C^{l\left(w^{'}\right)}\end{aligned}$$

But, by definition,

$$s\left(a^{'},b^{'}\right)=\sum_{w^{'}\in\mathcal{W}_{a^{'},b^{'}}}p\left(w^{'}\right)C^{l\left(w^{'}\right)}$$

This completes the proof that the expected f-meeting distances satisfy the recursive form of Equation 3.1. By the uniqueness result of Theorem 6, it follows that this is the same as the solution that can be arrived as a limit of the iterative form, thus establishing the equivalence of the Generalized Random Surfer-Pair model and its recursive form.

Bibliography

1. **Amsler, R. A.** (1972). Applications of citation-based automatic classification.

2. **Antonellis, I.**, **H. G. Molina**, and **C. C. Chang** (2008). Simrank++: Query rewriting through link analysis of the click graph. *Proc. VLDB Endow.*, **1**(1), 408–421. ISSN 2150-8097. URL http://dx.doi.org/10.14778/1453856.1453903.

3. **Brin, S.** and **L. Page**, The anatomy of a large-scale hypertextual web search engine. *In Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1998. URL http://dl.acm.org/citation.cfm?id=297805.297827.

4. **Fogaras, D.** and **B. Rácz**, Scaling link-based similarity search. *In Proceedings of the 14th International Conference on World Wide Web*, WWW '05. ACM, New York, NY, USA, 2005. ISBN 1-59593-046-9. URL http://doi.acm.org/10.1145/1060745.1060839.

5. **Hamedani, M. R.** and **S.-W. Kim**, Simrank and its variants in academic literature data: Measures and evaluation. *In Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-3739-7. URL http://doi.acm.org/10.1145/2851613.2851811.

6. **Haveliwala, T. H.**, Topic-sensitive pagerank. *In Proceedings of the 11th International Conference on World Wide Web*, WWW '02. ACM, New York, NY, USA, 2002. ISBN 1-58113-449-5. URL http://doi.acm.org/10.1145/511446.511513.

7. **Jeh, G.** and **J. Widom**, Simrank: A measure of structural-context similarity. *In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02. ACM, New York, NY, USA, 2002. ISBN 1-58113-567-X. URL http://doi.acm.org/10.1145/775047.775126.

8. **Jiang, W.**, **J. Vaidya**, **Z. Balaporia**, **C. Clifton**, and **B. Banich**, Knowledge discovery from transportation network data. *In Proceedings of the 21st International Conference on Data Engineering*, ICDE '05. IEEE Computer Society, Washington, DC, USA, 2005. ISBN 0-7695-2285-8. URL http://dx.doi.org/10.1109/ICDE.2005.82.

9. **Kessler, M. M.** (1963). Bibliographic coupling between scientific papers. *American Documentation*, **14**(1), 10–25. ISSN 1936-6108. URL http://dx.doi.org/10.1002/asi.5090140103.

10. **Kleinberg, J. M.** (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, **46**(5), 604–632. ISSN 0004-5411. URL http://doi.acm.org/10.1145/324133.324140.

11. **Küçüktunç, O.**, **E. Saule**, **K. Kaya**, and **Ü. V. Çatalyürek** (2012). Recommendation on academic networks using direction aware citation analysis. *CoRR*, **abs/1205.1143**. URL http://arxiv.org/abs/1205.1143.

12. **Lin, Z.**, **I. King**, and **M. R. Lyu**, Pagesim: A novel link-based similarity measure for the world wide web. *In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06. IEEE Computer Society, Washington, DC, USA, 2006. ISBN 0-7695-2747-7. URL http://dx.doi.org/10.1109/WI.2006.127.

13. **Lin, Z.**, **M. R. Lyu**, and **I. King** (2012). Matchsim: A novel similarity measure based on maximum neighborhood matching. *Knowl. Inf. Syst.*, **32**(1), 141–166. ISSN 0219-1377. URL http://dx.doi.org/10.1007/s10115-011-0427-z.

14. **Lizorkin, D.**, **P. Velikhov**, **M. Grinev**, and **D. Turdakov** (2010). Accuracy estimate and optimization techniques for simrank computation. *The VLDB Journal*, **19**(1), 45–66. ISSN 1066-8888. URL http://dx.doi.org/10.1007/s00778-009-0168-8.

15. **Manning, C. D.**, **P. Raghavan**, and **H. Schütze**, *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

16. **Mavroforakis, C.**, **M. Mathioudakis**, and **A. Gionis** (2015). Absorbing random-walk centrality: Theory and algorithms. *CoRR*, **abs/1509.02533**. URL http://arxiv.org/abs/1509.02533.

17. **Narwekar, A.** (2016). An academic search engine and problems in citation networks. *Dual Degree Thesis, Indian Institute of Technology Madras*.

18. **Rothe, S.** and **H. Schütze**, Cosimrank: A flexible & efficient graph-theoretic similarity measure. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 2014. URL http://www.aclweb.org/anthology/P14-1131.

19. **Russell, R. B.** and **P. Aloy** (2008). Targeting and tinkering with interaction networks. *Nat Chem Biol*, **4**(11), 666–673. ISSN 1552-4450. URL http://dx.doi.org/10.1038/nchembio.119.

20. **Singh, A. P.**, **A. Gunawardana**, **C. Meek**, and **A. C. Sudendran**, Recommendations using absorbing random walks. *In North East Student Colloquium on Artificial Intelligence*. 2007. URL https://www.microsoft.com/en-us/research/publication/recommendations

21. **Small, H.** (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, **24**(4), 265–269. ISSN 1097-4571. URL http://dx.doi.org/10.1002/asi.4630240406.

22. **Tang, J.**, **J. Zhang**, **L. Yao**, **J. Li**, **L. Zhang**, and **Z. Su**, Arnetminer: Extraction and mining of academic social networks. *In Proceedings of the 14th*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-193-4. URL http://doi.acm.org/10.1145/1401890.1402008.

23. **Yin, X.**, **J. Han**, and **P. S. Yu**, Linkclus: Efficient clustering via heterogeneous semantic links. *In Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06. VLDB Endowment, 2006. URL http://dl.acm.org/citation.cfm?id=1182635.1164165.

24. **Yoon, S.-H.**, **S.-W. Kim**, and **S. Park** (2016). C-rank. *Inf. Sci.*, **326**(C), 25–40. ISSN 0020-0255. URL http://dx.doi.org/10.1016/j.ins.2015.07.036.

25. **Yu, W.**, **X. Lin**, **W. Zhang**, **L. Chang**, and **J. Pei** (2013). More is simpler: Effectively and efficiently assessing node-pair similarities based on hyperlinks. *Proc. VLDB Endow.*, **7**(1), 13–24. ISSN 2150-8097. URL http://dx.doi.org/10.14778/2732219.2732221.

26. **Zhao, P.**, **J. Han**, and **Y. Sun**, P-rank: A comprehensive structural similarity measure over information networks. *In Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-512-3. URL http://doi.acm.org/10.1145/1645953.1646025.