The Chinese University of Hong Kong
Department of Computer Science and Engineering

Ph.D. -- Term Paper

Cover Sheet

Title: PageSim: A Novel Link-based Measure of Web Page Similarity

Name: Zhenjiang Lin

Student I.D.: 05267230

Contact Tel. No.: 67056279 Email A/C: zjlin@cse.cuhk.e

Supervisor(s): Michael R. Lyu and Irwin King

Marker(s):

Mode of Study: ☐ Part-time ☑ Full-time

Submission Date:

Term: Algorithms, Measurement

Field(s): Information Search and Retrieval, Graph Algorithms

Presentation Date:

Time:

Venue:

Abstract

To find similar web pages to a query page on the Web, this paper introduces a novel link-based similarity measure, called *PageSim*. Contrast to SimRank, a recursive refinement of cocitation, PageSim can measure similarity between *any* two web pages, whereas SimRank cannot in some cases. We give some intuitions to the PageSim model, and outline the model with mathematical definitions. We also suggest techniques for efficient computation of PageSim scores. Finally, we give an example to illustrate its effectiveness.

Keywords: similarity measure, link analysis, search engine, PageRank, SimRank

1 Introduction

Finding similar web pages to a query page is a crucial task for a search engine. Recently, a variety of link-based similarity measures, which use only the hyperlinks in the Web, have been proposed for this task. This includes companion algorithm [DH99], cocitation algorithm [DH99], and SimRank [JW02], etc.

In this paper, we propose a novel link-based similarity measure, called PageSim. Contrast to SimRank, our method can measure similarity between any two web pages, whereas SimRank cannot in some cases.

SimRank is a fixed point of the recursive definition: *two pages are similar if they are linked to by similar pages*. Numerically, for any web page u and v, this is specified by defining $simrank(u, u) = 1$ and

$$simrank(u, v) = \gamma \cdot \frac{\sum_{a \in I(u)} \sum_{b \in I(v)} simrank(a, b)}{|I(u)||I(v)|} \tag{1}$$

for $u \neq v$ and $\gamma \in (0, 1)$, where $I(x)$ denotes the set of inlink pages of x, $|I(x)|$ denotes the cardinality of the set. If $I(u)$ or $I(v)$ is empty, then $simrank(u, v)$ is zero by definition. The SimRank iteration starts with $simrank_0(u, v) = 1$ for $u = v$ and $simrank_0(u, v) = 0$ for $u \neq v$. The *SimRank score* between u and v is defined as $lim_{k \to \infty} simrank_k(u, v)$.

Unfortunately, the result of SimRank is not convincing in some cases. In one case, if one of two web pages has no inlink, then the SimRank score of them is zero by definition, which means they are not similar. However, this

is not always true. For example, in Figure 2 of section 5, v_1 has no inlink, but it is clear that both v_2 and v_3 have some similarity with it for they are linked to by v_1 . In another case (also in Figure 2), SimRank concludes that v_2 and v_4 are not similar. In fact, obviously v_2 and v_4 indeed have some similarity, for they *link to each other*. More detailed illustration is given in the last part of this paper.

The content of this paper is organized as follows. In next section, we related work in similarity measures. Section 3 gives some intuitions to the PageSim model. The mathematical definitions of PageSim model and analysis on PageSim algorithm is presented in section 4. Evaluation on PageSim and experiments on propagation radius are given in section 5 and section 6 respectively. The conclusion and future work are given as the last part of this paper.

2 Related Work

The problem of finding *related* or *similar* pages to a query page on the Web arises in a variety of web applications, such as web search engines and web document classification. On the other hand, it shows that users of the Web usually examine only the first few pages of search results. Therefore, it is natural for these web applications to require effective similarity measures to rank similarity between web pages, on either the textual content of pages or the hyperlink structure of the Web, in order to provide users the results most fit their desire.

Measuring the "similarity" between objects is required in many applications, and numbers of domain-specific similarity measures have been developed. For example, measures that based on *matching text* may be used to find similar documents to a given query in a document corpus. And for collaborative filtering in a recommender system [GNOT92, KMM+97, SM95], similar users may be grouped by *users' preferences*. In particular, several link-based similarity search algorithms were suggested to exploit the similarity information hidden in the link structure of graph, such as SimRank [JW02], cocitation algorithm [DH99], and companion algorithm [DH99], etc [LNK03]. Further methods arise from graph theory, such as similarity measure that based on network flows [LJMJ01].

Because link structure is more resistant to spamming than textual content [AAR01], this paper focuses only on the link-based similarity measures which

computing similarities solely from the hyperlink structure modeled by the web graph, with vertices corresponding to web pages and directed arcs to the hyperlinks between pages.

3 Intuitions Behind PageSim

3.1 Web Graph Model

We model the Web as a directed graph $G = (V, E)$ with vertices V representing web pages $v_i (i = 1, 2, \cdots, n = |E|)$ and directed edges E representing hyperlinks between web pages.

Definition 1 *Let $I(v)$ denotes the set of inlink pages of v and $O(v)$ denotes the set of outlink pages of v, for $v \in V$.*

Definition 2 *Let $path(u_1, u_s)$ denotes a sequence of vertices $u_1, u_2, \ldots, u_s$ such that $(u_i, u_{i+1}) \in E$ $(i = 1, \cdots, s-1)$ and u_i are distinct, it is called a **path** from u to v.*

Definition 3 *Let $length(p)$ denotes the **length** of path p, define $length(p) = |p| - 1$.*

Definition 4 *Let $PATH(u, v)$ denotes the set of all possible paths from page u to v.*

3.2 PageRank

PageRank is a well known ranking algorithm which uses only link information to assign global importance scores to all pages on the Web. Because our proposed algorithm rely on PageRank, we offer a short overview in this section.

PageRank was introduced by Page and Brin [PBMW98]. The intuition behind the algorithm is

"*a page has high rank if the sum of the ranks of its backlinks is high.*"

It assumes that the number of incoming links to a page is related to that page's popularity among average web users (people would point to pages

that they find important). Correspondingly, PageRank is based on a mutual reinforcement between pages. From the viewpoint of random walk, the PageRank score of a web page can be considered as the possibility that this web page is being visited by a random web surfer at a certain time.

The PageRank score of web pages can be computed using the following recursive algorithm:

$$\mathbf{X}(t+1) = dW\mathbf{X}(t) + (1-d)\mathbf{I}_n, \tag{2}$$

where $\mathbf{X} \in \mathcal{R}^n$ is an n-dimensional vector denoting the PageRank of web pages. $\mathbf{X}(t)$ denotes the PageRank vector at the t-th iteration. $W = (w_{ij})_{n\times n}$ is the *transition matrix*:

$$w_{ij} = \begin{cases} \frac{1}{|O(v_j)|} & (v_j, v_i) \in V, \\ \\ 0 & otherwise. \end{cases}$$

$\mathbf{I}_n$ is an n-dimensional vector with all elements equal to 1. d is a damping factor. The PageRank of total n web pages is given by the steady state solution of (2).

PageRank appears to *resist spamming* because all reputation stems from the initial votes. In [Cla04], the authors points out that the cost of acquiring a PageRank r is $rc(P)$, where $c(P)$ is the total money spent by all web sites on domain names and IP addresses. This means, although PageRank can clearly be manipulated, the cost is expensive. This helps put top search placements out of reach of spammers. By now, Google, which based on PageRank algorithm, is the most popular search engine for its robust against spamming.

3.3 PageSim

PageSim can be considered as an extension of cocitation algorithm, in which the similarity score between two web pages is defined by the number of inlink neighbors that they have in common. Actually, on the Web, not all links are equally important. For example, if the only common neighbor of page a and b is the Yahoo home page [yah], whereas page a and c have several common neighbors from obscure places, then which page is more similar to page a, page b or page c? As we know, hyperlink from web page u to v can be considered as a recommendation of page v by page u [AAR01], and the more

important a web page is, the more important its recommendation is. Evidently, the reasonable answer should be page b, since the Yahoo home page is much more "important". In another perspective, the action of recommendation can be considered that page u *propagates* some kind of similarity to page v, and the more pages it links to, the less similarity it should propagate to each of these pages. Therefore, it is also reasonable to think that the Yahoo home page has some kind of similarity with both page a and page b.

Since PageRank is one of the most prominent ranking algorithm which assigns global ranking scores to all pages on the Web, we take the PageRank score of a web page as the *importance* (*weight* or *similarity score*) of it in the PageSim method. The intuitions to PageSim model is described as follows, and the mathematical definitions will be given later.

At the beginning, each web page only contains its own similarity score, and then each web page propagates its own similarity score to its outlink neighbors, receiving and propagating the similarity scores of others at the same time. After the propagation, each page contains its own similarity score as well as the similarity scores of others. These scores are stored in a vector called the *similarity vector* of this page. Then we can calculate the **PageSim score** of each pair of pages by *summing their common similarity scores up.*

4 The PageSim Algorithm

4.1 Definitions

Before giving the detailed description of PageSim algorithm, we introduce some important definitions first.

Definition 5 *Let $PR(v)$ denotes the PageRank score of page v, for $v \in V$.*

Definition 6 *Let $PG(u, v)$ denotes the PageRank score that page u propagates to page v through $PATH(u, v)$, that is,*

$$PG(u, v) = \sum_{p \in PATH(u,v)} \frac{PR(u)}{\prod_{w \in p, w \neq v} |O(w)|}, \tag{3}$$

where $u, v \in V$.

Definition 7 *Let* $\overrightarrow{PS}(v)$ *denotes the* ***similarity vector*** *of page* v*, we have*

$$\overrightarrow{PS}(v) = (PG(v_i, v))^T, i = 1, \cdots, n,$$

where $v, v_i \in V$*.*

Definition 8 *Let* $PS(u, v)$ *denotes the* ***PageSim score*** *of page* u *and page* v*,*

$$PS(u, v) = \sum_{i=1}^{n} min(PG(v_i, u), PG(v_i, v))$$

where $u, v \in V$*.*

4.2 PageSim Algorithm

The PageRank propagation process is very much like a *depth-first traversal* (DFT), but there is a slight difference between them: in PageSim, PageRank score is propagated along "paths" rather than along "branches" in DFT (see the definition of PG given in section 4.1). However, we can implement this process using a DFT-like algorithm. For better understanding of the propagation process, we give an simple illustration below.

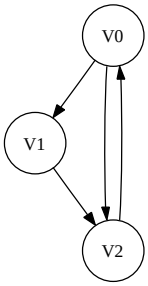

Figure 1: PageRank propagation process

In Figure 1, suppose $PR(v_0) = 1$. The PageRank score propagation process of page v_0's score (we refer "score" to "PageRank score" in this illustration) is described as follows:

*path*1 v_0 propagates 1/2 score to v_1, then v_1 propagates 1/2 score to v_2. v_2 does not propagate score to v_0 because v_0 is already in this path, therefore the propagation along this path ends;

*path*2 v_0 propagates 1/2 score to v_2, same reason as in "*path*1", the propagation along this path ends at v_2.

Therefore, $PG(v_0, v_0) = PG(v_0, v_2) = 1$ and $PG(v_0, v_1) = 1/2$. The obtained result implies that v_2 is more similar to v_0 than v_1, although the whole propagation process is unfinished.

Let's look deeper insight into PageSim. Let k be the average number of one web page's outlinks, i.e., $k = \sum_{i=1}^{n} |O(v_i)|$. The computational complexity of propagating one page's similarity to all the others is $O(k^n)$, which is too high. On the other hand, from the definition of PG, we have

$$EPG(u, v, score) = \frac{score}{k^L} \tag{4}$$

where $EPG(u, v, score)$ denotes the expectation of PageRank score that propagated from page u to v along one path $path(u, v)$, and $L = length(path(u, v))$. This means the PageRank score that propagated to *distant* pages drops very quickly if $k \geq 2$ holds (which is certainly true).

Since PageRank scores propagated to distant pages is very small and contribute very little to the summation, it is reasonable to limit the radius of propagation, which can be regarded as a tradeoff between efficiency and precision. This technique is called *pruning*, that is, we *prune* the tiny PageRank score propagation processes to reduce the resource requirements. By this way, the complexity drops to $O(k^r)$, where $r \in \mathcal{R}$ is the radius of propagation, i.e., the maximum length of propagation path.

Now, we can conclude the complexity of PageSim. Because k is likely to be much less than n, we can think of the approximate PageSim algorithm as being linear with a possibly large constant factor, that is, the time complexity of propagation procedure is $O(Cn)$. However, the time complexity of computing PageSim scores between all $O(n^2)$ pairs of pages is $O(n^3)$, therefore, the total time complexity of PageSim is $O(n^3)$. The space complexity is $O(n^2)$ since each web page have to contain a n-dimensional *similarity vector*.

The detailed PageSim algorithm is given below.

PageSim Algorithm
Input
G : web graph $G(V, E)$ with known PageRank scores,
r : propagation radius.
Output

$PSmatrix$:PageSim matrix $(PS(v_i, v_j))_{n\times n}, i, j = 1, \cdots, n$.

1: **procedure** PageSim(G,r)
2: **for** $i \leftarrow 1, n$ **do**
3: **call** $PR\_Prop(G,r,v_i)$ ▷ propagate PageRank of v_i
4: **end for**
5: **for** $i \leftarrow 1, n$ **do**
6: **for** $j \leftarrow 1, n$ **do**
7: calculate $PS(v_i, v_j)$
8: **end for**
9: **end for**
10: **end procedure**

The **PR_Prop** procedure for propagating PageRank score of web pages is also given below, which is a "path-based DFT-like" algorithm.

PageRank Propagation Algorithm
Input
G : web graph $G(V, E)$ with known PageRank scores,
r : propagation radius,
v : a web page whose PageRank is to be propagated.
Output
G : the graph G after the PageRank propagation of v.
Data Structure
path_stack : a stack used to store pages on current path.
max_path_len : the size of *path_stack*.
path_len: the length of current path.

1: **procedure** PR_Prop(G,r,v)
2: $v^* \leftarrow v$
3: **if** v^* has no outlinks **then**
4: **return**
5: **end if**
6: $v^*.PR\_prop \leftarrow \frac{v^*.PR}{v^*.outlink\_num}$
▷ compute the PageRank score that
▷ v^* propagates to its outlink pages
7: $path\_len \leftarrow 0$
8: **while** $v^* \neq NULL$ **do**
9: set v^* is visited

```
    visit_next ← FALSE
    if remains outlink pages of v* which unvisited then
        if path_len < max_path_len then
            v_o ← one unvisited outlink page of v*
            add v*.PR_prop to v_o's similarity vector
            if v_o has no outlinks then
                v_o.PR_prop ← v*.PR_prop / v*.outlink_num
            else
                v_o.PR_prop ← 0
            end if
            PUSH(v*, path_stack)
            path_len ← path_len + 1
            v* ← v_o
            visit_next ← TRUE
        end if
    end if
    if visit_next = FALSE then
        set v* is unvisited
        v* ← POP(path_stack)
        path_len ← path_len − 1
    end if
end while
end procedure
```

4.3 Properties of PageSim

We list Some other properties of PageSim below, which can be easily deduced from the definitions in section 4.1.

1. The PageSim scores are *symmetric*, i.e.,

$$PageSim(u, v) = PageSim(v, u);$$

2. Each page is the most similar page to itself, i.e.,

$$PageSim(u, u) = \max_{v \in V} PageSim(u, v)$$

5 PageSim *vs* SimRank

A good evaluation of PageSim is difficult without performing extensive user studies or having a reliable external measure of similarity to compare against. In this section, we give a simple example in which PageSim is compared with SimRank to illustrate the performance of PageSim.

For a given graph $G(V, E)$, where $V = \{v_i\}(i = 1, \cdots, 6)$ (see Figure 2). Let $\overrightarrow{PR}(V) = (PR(v_i))^T, i = 1, \cdots, 6$. We have

$$\overrightarrow{PR}(V) = (0.08, 0.23, 0.18, 0.14, 0.14, 0.23)^T.$$

The *PageSim score matrix* is

$$\begin{pmatrix} 0.08 & 0.04 & 0.05 & 0.01 & 0.01 & 0.05 \\ 0.04 & 0.41 & 0.16 & 0.23 & 0.14 & 0.16 \\ 0.05 & 0.16 & 0.35 & 0.14 & 0.14 & 0.35 \\ 0.01 & 0.23 & 0.14 & 0.23 & 0.14 & 0.14 \\ 0.01 & 0.14 & 0.14 & 0.14 & 0.28 & 0.14 \\ 0.05 & 0.16 & 0.35 & 0.14 & 0.14 & 0.58 \end{pmatrix}.$$

Let $top(v, t)$ denotes the top t similar pages to page v (excluding v). Let $t = 2$, we have

$$\begin{array}{ll} top(v_1, 2) = \{v_3, v_6\}, & top(v_2, 2) = \{v_4, v_{3,6}\}, \\ top(v_3, 2) = \{v_6, v_2\}, & top(v_4, 2) = \{v_2, v_{3,5,6}\}, \\ top(v_5, 2) = \{v_{2,3,4,6}, v_1\}, & top(v_6, 2) = \{v_3, v_2\}. \end{array}$$

The *SimRank score matrix* of graph G is

$$\begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.25 & 0.00 & 0.00 & 0.25 \\ 0.00 & 0.25 & 1.00 & 0.50 & 0.50 & 0.13 \\ 0.00 & 0.00 & 0.50 & 1.00 & 1.00 & 0.25 \\ 0.00 & 0.00 & 0.50 & 1.00 & 1.00 & 0.25 \\ 0.00 & 0.25 & 0.13 & 0.25 & 0.25 & 1.00 \end{pmatrix}.$$

Thus, we have

$$\begin{array}{ll} top(v_1, 2) = \{\}, & top(v_2, 2) = \{v_3, v_6\}, \\ top(v_3, 2) = \{v_{4,5}, v_2\}, & top(v_4, 2) = \{v_5, v_3\}, \\ top(v_5, 2) = \{v_4, v_3\}, & top(v_6, 2) = \{v_{2,4,5}, v_3\}. \end{array}$$

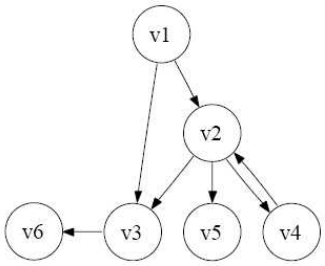

Figure 2: graph G

We can see that the results of PageSim and SimRank are different. First, SimRank shows that there's no page similar to v_1. While PageSim shows that v_3 is most similar to v_1, which is more reasonable. Because the fact that v_1 links to v_3 implies v_1 "considers" v_3 has some level of similarity with it. Secondly, SimRank shows v_4 is not similar to v_2, while PageSim shows that is not true. Obviously, v_2 and v_4 are similar, for they *link to each other*. Moreover, PageSim considers that v_4 is most similar to v_2. SimRank shows v_3 is most similar to v_2, for they have a common inlink page v_1. We believe PageSim is the winner in this situation because the "link to each other" relationship really implies stronger similarity than that of the "common inlink" relationship.

6 Experiments on Propagation Radius

In section 4.2, we proposed the *pruning technique* which reduce the time complexity of propagation to $O(k^r n)$, where r is the propagation radius. In this section, we want to get empirical results on the radius r through experiments.

6.1 Data Sets

We ran experiments on two kind of data sets: *real web graph* and *synthetic graph*.

real web graph Web graphs crawled from *cuhk.edu.hk*.

synthetic graph Randomly generated graphs according to the power law.

The degree sequences of the Web are shown to be well approximated by a power law distribution [KKR+99, KRRT99a, KRRT99b], that is, the probability that a Web page has k outlinks (inlinks) follows a power:

$$P_{out}(k) \sim k^{-\gamma_{out}} (P_{in}(k) \sim k^{-\gamma_{in}}).$$

Therefore, we can generate synthetic Web-like random graphs to be the inputs of our algorithm. Several approaches to modeling power law graphs [BAJ99, KKR+99, KRRT99b] have been proposed. In our experiments, we use the (α, β) model [KRRT99b] to generate random graphs. By setting $\alpha = 0.52$ and $\beta = 0.58$, the model generates a random graph following power law with $\gamma_{out} = 2.1$ and $\gamma_{in} = 2.38$, both of these values match the Web.

6.2 Methodology

Given a web graph G and propagation radius r, let PS_r denotes the PageSim matrix produced by $PageSim(G, r)$, where $r > 0$. Define $PS_0 = 0$.

In our experiments, we check the difference between PS_r and PS_{r-1}, which denoted by $Diff(r)$. Simply, we define

$$Diff(r) = \|PS_r - PS_{r-1}\|_2,$$

where $\|\cdot\|_2$ is Euclidean norm. Apparently, we may think that $Diff(r)$ should approaches to zero as r approaches to $n-1$, i.e.,

$$Diff(r) \to 0, r \to n-1. \tag{5}$$

Therefore, the goals of our experiments is:

1. to check the correctness of our hypothesis in (5).
2. if (5) is true, then can we get an empirical propagation radius r through experiments?

6.3 Experimental Results

We run PageSim algorithm on two kinds of data set: *real web graph* and *synthetic graph*, both of them consist of three graphs which contain approximately 500, 5000, and 8000 vertices respectively. The curves are shown in

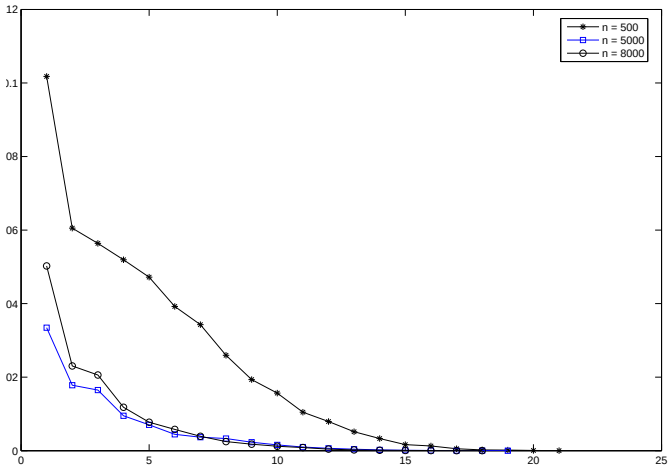

Figure 3: real web graph

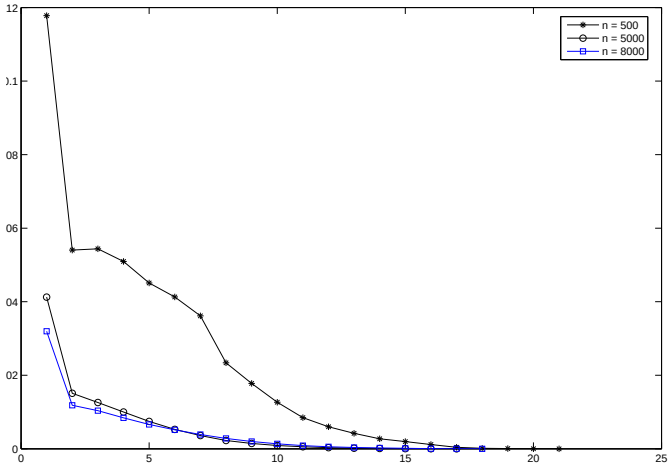

Figure 4: synthetic graph

Figure 3 and Figure 4, with horizon axes representing r, and vertical axes representing $Diff(r)$.

In both Figures , the curves drops quickly as the propagation radius r increases, which shows that our hypothesis is correct. Generally, $Diff(r) \approx 0$ when $r \approx 10$. This means the PageRank scores that propagated to the web pages more than 10 hops long is small enough to be omitted. Therefore, empirically we can choose $r = 10$ to be the PageRank propagation radius in practice to improve the efficiency of PageSim.

7 Conclusion and Future Work

This paper introduces PageSim, a novel link-based similarity measure. Based on the strategy of *PageRank score propagation*, PageSim is capable of measuring similarity between any two web pages.

There are numbers of avenues for future work. Foremost, we must address efficiency and scalability issues. Although we reduced the time complexity of PageRank propagation procedure to $O(k^r n)$, and concluded the empirical value of propagation radius through experiments, however, the time complexity of computing PageSim scores of web pages is $O(n^3)$, which result in the high time complexity of PageSim. Moreover, due to the small size of the input graphs, our results may be inapplicable to huge web graphs which include millions of thousands of web pages.

On the other hand, The storage required by PageSim is $O(n^2)$, which is also unacceptable for web applications. Since averagely a web page has k inlinks, and the propagation radius is limited, there will be lots of zeros in *similarity vectors*. One possible direction is to use new data structure which storing only received PageRank score instead storing all possible PageRank score.

A second area of future work is to make PageSim more accurate. Notice that if we modify the definition of PG to be

$$PG(u, v) = \sum_{p \in PATH(u,v)} \frac{PR(u)}{\prod_{w \in p, w \neq v} c(p, w)|O(w)|},$$

where $c(p, w) \in (0, 1]$ is a *decay factor function*, we may possibly obtain a more precise result. We also leave it to our future research.

8 Acknowledgments

This work is supported by grants from the Research Grants Councils of the HKSAR, China (Project No. CUHK4205/04E and Project No. CUHK4235 /04E) and is affiliated with the VIEW Technologies Laboratory and the Microsoft-CUHK Joint Laboratory for Human-centric Computing & Interface Technologies.

References

[AAR01] Hector Garcia-Molina Andreas Paepcke Arvind Arasu, Junghoo Cho and Sriram Raghavan. Searching the web. *ACM Trans. Inter. Tech.*, 1(1):2–43, 2001.

[BAJ99] A. Barab'asi, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world wide web, 1999.

[Cla04] Andrew Clausen. The cost of attack of pagerank. In *In Proc. of the International Conference on Agents, Web Technologies and Internet Commerce (IAWTIC), Gold Coast,*, 2004.

[DH99] Jeffrey Dean and Monika R. Henzinger. Finding related pages in the World Wide Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1467–1479, 1999.

[GNOT92] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[JW02] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. *Proc. of SIGKDD*, 2002.

[KKR+99] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–??, 1999.

[KMM+97] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: applying

collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, 1997.

[KRRT99a] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, 1999.

[KRRT99b] S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Extracting large-scale knowledge bases from the web. In *The VLDB Journal*, pages 639–650, 1999.

[LJMJ01] Wangzhong Lu, Jeannette Janssen, Evangelos Milios, and Nathalie Japkowicz. Node similarity in networked information spaces. In *CASCON '01: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, page 11. IBM Press, 2001.

[LNK03] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Twelfth International Conference on Information and Knowledge Management*, pages 556–559. ACM, November 2003.

[PBMW98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[SM95] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating word of mouth. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[yah] http://www.yahoo.com.