PageSim: A Novel Link-based Similarity Measure for the World Wide Web

Zhenjiang Lin, Irwin King, and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
{zjlin, lyu, king}@cse.cuhk.edu.hk

Abstract

The requirement for measuring the similarity between web pages arises in many applications on the Web, such as web searching engine and web document classification. According to the unique characteristics of the Web, which are huge, rapidly growing, high dynamic, and untrustworthy, we propose a novel link-based similarity measure called PageSim. Based on the strategy of PageRank score propagation, PageSim is efficient, scalable, stable, and "fairly" robust, and therefore is applicable to the Web. We present intuitions behind the PageSim model, and outline the model with mathematical definitions. We also suggest the pruning technique for efficient computation of PageSim scores, and conduct experiments to illustrate the effectiveness and specialities of PageSim.

1. Introduction

Unlike keyword searching which takes a user-formulated query as input and produces a set of *relevant* information, similarity searching, or searching by instance, takes an instance as input and produces a set of *similar* instances.

The World Wide Web (or simply "the Web") is a global, distributed, read-write information space. Many web applications in both scientific and business domains require efficient similarity measure to extract useful knowledge from the Web. For example, the "related pages" service of web search engines searches for *related* or *similar* web pages to a query web page. Web document classification is another important web application which organizes web pages into a hierarchical structure according to the degree of similarity between web pages. Another important web application is identifying *web community* which is a collection of web pages sharing a common topic [4, 13].

The Web provides people increasingly huge volume of information of various domains, and makes mining tasks on the Web more and more difficulty at the same time. Although the Web can be modeled by a graph, with vertices corresponding to web pages and directed arcs to the hyperlinks between pages, there are several characteristics distinguish the web graph from ordinary graph.

1. **Huge**: Undoubtedly, with billions of web pages created by millions of web page authors, the Web has became a tremendously rich information warehouse.

2. **Rapidly Growing**: Most studies agree that the Web grows at an exponential rate [10, 16], which has been estimated to be roughly one million pages per day [10].

3. **High Dynamic**: Unlike books in a traditional library, web pages continue to change after they are initially created and indexed by search engines [3]. According to [16], basically there are two dimensions of web dynamics: *growth dynamic* and *update dynamic*. The former indicates that the Web grows in size, the latter indicates that both the content and the link structure of the Web are constantly updated.

4. **Untrustworthy**: It is well known that the Web is an untrustworthy world due to the fact that its contents, including textual content of web pages and hyperlinks between web pages, are prone to be manipulated, or *spammed*. *Spammers* on the Web use various techniques to "mislead search engines and give some pages higher ranking than they deserve" [5], this action is called *web spamming* [5]. Experts consider web spamming the single most difficult challenge web searching is facing today [6].

According to the above unique characteristics of the Web, naturally there are four corresponding requirements for the algorithms on the Web.

1. **Efficiency**: Evidently, only those algorithms with low time and space complexity are applicable to the huge Web. Even $O(n^2)$ is rather high for web applications.

2. **Scalability**: The dramatic growth rate of the Web poses a serious challenge of scalability for web applications that aspire to cover a large part of the Web [16].

3. **Stability**: An algorithm should be stable to perturbations of the Web, including link structure and content of web pages.

4. **Robustness**: We use the term “robust” to indicate that an algorithm on the Web is resistent to commonly used web spamming techniques.

The above requirements motivate the work in this paper, which results in a novel link-based similarity measure called PageSim. Based on *PageRank score propagation* strategy, PageSim is an efficient, scalable, Stable, and “fairly” robust similarity measure for the Web. We use the adverb “fairly” to emphasis that PageSim is not promised to be resistant to *any* web spamming techniques. Actually, it is impossible for any algorithm on the Web to make such a statement. However, it is possible for a well-designed algorithm to be “fairly” robust by increasing the cost of spamming attack. One of the successful examples is PageRank [15], the fundamental algorithm of Google [1]. Web spamming has became such an important issue on the Web, whereas most existing similarity measures have not taken it into account.

The main contributions of this paper are as follows.

1. A novel link-based algorithm, PageSim, is proposed to measure the similarity between web pages.

2. The low time and space complexity and the parallelism property make PageSim especially suitable for distributed computing.

3. A rigorous mathematical definition for PageSim similarity scoring, and an algorithm to compute PageSim are presented.

4. Extensive experiments are conducted to illustrate the effectiveness and specialities of PageSim.

2. Related Work

The problem of finding similar objects has been studied extensively in the field of information retrieval and recommender systems for many years, and a variety of text-based similarity measures have been proposed, such as the *cosine similarity* and the *TFIDF* (Term Frequency-Inverse Document Frequency) model [17]. A problem of the text-based methods is that they generally require large storage and long computing time due to the need of full-text comparison. Moreover, they are prone to be manipulated by keyword spamming.

The link-based similarity measures were first developed in the bibliometrics field which studies the citation patterns of scientific papers (or other publications), and infers relationships between papers from their cross-citations [9, 18]. Two noteworthy methods are *co-citation* and *bibliographic coupling*. Co-citation measures similarity between two papers based on the number of papers which cite both of them. In bibliographic coupling, similarity is based on the number of papers cited by both of the two papers.

Several link-based similarity measures have been proposed in the past few years. The *Maximum Flow/Minimum Cut* was proposed for measuring the similarity of scientific papers in a citation graph [12]. The SimRank algorithm was proposed to measure similarity of the structural context. It is a recursive refinement of co-citation based on the assumption that “two objects are similar if they are linked to similar objects”. Furthermore, link-based similarity measures have been suggested over the web graph. We refer to [11], which contains an exhaustive list of link-based similarity functions.

3. Intuitions Behind PageSim

Research have revealed that the vast link structure of the Web is an indicator of an individual page’s importance. This section presents the key idea of PageSim which uses only the link structure of the Web.

3.1. More About PageRank

PageRank is a well known ranking algorithm which uses only link information to assign global importance scores to all pages on the Web. The intuition behind the algorithm is “*a page has high rank if the sum of the ranks of its backlinks (in-links) is high*.” The PageRank score of web pages can be computed using the following recursive algorithm:

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

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, and d is a damping factor. The PageRank of total n web pages is given by the steady state solution of equation (1).

3.2. PageSim: PageRank Score Propagation

PageSim can be considered as an extension of Co-citation algorithm, in which the similarity score between two web pages is defined by the number of in-link neighbors that they have in common. Actually, on the Web, not all links are equally important. For example, if the only common in-link of page a and b comes from Yahoo's home page , whereas page a and c have several common in-links 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* [2], and the more important a web page is, the more important its recommendation is. Obviously, the reasonable answer should be page b, since Yahoo's home page is much more important.

On the other hand, the action of recommendation can be considered that page *u propagates* some of its unique "feature information" to page *v* through hyperlinks. Therefore, both page a and page b receive part of feature information from Yahoo's home page, which implies all of these three web pages are similar web pages to one another.

Since PageRank is one of the most prominent ranking algorithm which assigns global *importance* scores to all web pages, PageSim adopts the PageRank (PR) score to represent the value of feature information of a web page. Another important reason we choose PR scores is that PageSim can benefit from the advantages of PageRank, which is stability and robustness.

The intuitions behind PageSim model is described as follows. Initially, each web page only contains its own feature information (PR score). When the propagation process begin, each web page start to propagate its feature information to all of its out-link neighbors through hyperlinks, receiving and propagating the feature information of others at the same time. After the propagation process finish, each page contains its own feature information as well as others'; all of these feature information are stored in a vector called the *feature vector* of this page. Then we can calculate the *PageSim score* of each pair of pages by "comparing their common feature information".

4. Mathematical Model of PageSim

In this section we give the formal mathematical definitions of PageSim and give a simple example to illustrate the process of PageRank score propagation.

4.1. Web Graph Model

We model the Web as a directed graph $G = (V, E)$ with vertices V representing web pages $v_i(i = 1, 2, \cdots, n)$ and directed edges E representing hyperlinks between pages. Let $I(v)$ denote the set of in-link neighbors of page v and $O(v)$ denote the set of out-link neighbors of page v.

Definition 1 *Let $path(u_1, u_s)$ denote 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_1 to u_s.*

Definition 2 *Let $length(p)$ denote the **length** of path p, and define $length(p) = |p| - 1$. $|p|$ is the number of vertices in path p.*

Definition 3 *Let $PATH(u, v)$ denote the set of all possible paths from page u to v.*

4.2. Mathematical Definitions of PageSim

The mathematical definitions of PageSim are presented below, and the interpretations are given in section 4.3.

Definition 4 *Let $PR(v)$ denote the PR score of page v. Let $PG(u, v)$ denote the PR score of page u that propagated to page v through $PATH(u, v)$. We define*

$$PG(u,v) = \begin{cases} \sum_{p \in PATH(u,v)} \frac{d \cdot PR(u)}{\prod_{w \in p, w \neq v} |O(w)|}, & v \neq u, \\ PR(u) & v = u, \end{cases} \tag{2}$$

where $d \in (0, 1]$ is a decay factor and $u, v \in V$.

Definition 5 *Let $\overrightarrow{FV}(v)$ denote the **Feature Vector** of page v, we have*

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

where $v, v_i \in V$.

Definition 6 *Let $PS(u, v)$ denote the **PageSim score** between page u and page v. We define*

$$PS(u,v) = \sum_{i=1}^{n} \frac{min(PG(v_i,u), PG(v_i,v))^2}{max(PG(v_i,u), PG(v_i,v))}, \tag{3}$$

where $u, v \in V$.

4.3. PageSim Algorithm

For better understanding of the definitions in the previous subsection, we give more interpretations. There are two stages in PageSim algorithm: *PR score propagation stage* and *PageSim score computation stage*. Equation (2) and (3) correspond to the processes in the two stages respectively.

PR Score Propagation Stage The PR score propagation process is very much like the *depth-first traversal* (*DFT*) on a directed graph, except for a little difference between them. In PageSim, PR scores are propagated along "paths" (refer to the definition of *PG* given by equation (4)) rather than "branches" in DFT. That is to say, in the propagation process of web page *u*'s PR score, its PR score may be propagated to page *v* along different "paths" and accumulate there. Therefore, a web page may be visited several times in the propagation process of one web page, rather than only one time in the DFT algorithm. However, we can implement the propagation process in PageSim by constructing a DFT-like algorithm. We give a simple example below to illustrate the process of PR score propagation.

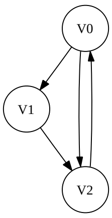

Figure 1. PR score propagation

In Figure 1, set $d = 0.8$ and suppose $PR(v_0) = 1$. The propagation process of page v_0's PR score is as follows.

*path*1 v_0 propagates 0.4 ($0.8 \times 1/2$) score to v_1, then v_1 propagates 0.32 ($0.8 \times 0.4/1$) to v_2. v_2 will not propagate the PR score to v_0, because v_0 is already in this path. Therefore, the propagation along this path ends;

*path*2 v_0 propagates 0.4 to v_2. Same reason as in "*path*1", the propagation along this path ends at v_2.

Therefore, we get $PR(v_0, v_0) = 1$, $PR(v_0, v_1) = 0.4$, and $PR(v_0, v_2) = 0.32 + 0.4 = 0.72$. These results imply that v_2 seems more similar to v_0 than v_1, although the whole propagation process is not finished.

PageSim Score Computation Stage In the computation stage, perhaps the simplest way to compute the similarity score of two web pages is summing their common feature information up. But this paper also takes the ratio of common feature information into account. Intuitively, closer the common feature information of two web pages are, more similar they are. Therefore we get the equation (3).

The PR score propagation process of a web page is encapsulated in the *PR_prop* sub-function, and the calculation of PageSim score between two pages is in the *PS_calc* sub-function. Since these sub-functions are rather straightforward and due to space constraints, we omit them to make the paper tidy.

5. Analysis of PageSim

In this section, we look insight into PageSim. First, the *pruning technique* is adopted by PageSim to reduce the resource requirements. After that, we give the time and space complexities of the algorithm.

5.1. Pruning Technique

Let k be the average number of one web page's outlinks, i.e., $k = (\sum_{i=1}^{n} |O(v_i)|)/n$. The time complexity of *PR_Prop* is $O(k^n)$, which is too high. From the definition of PG, we can easily deduce that

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

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

Since PR scores propagated to distant pages is so tiny that their contributions to the summation are tiny too, it is reasonable to increase the efficiency of PageSim by *pruning* the radius of propagation. This technique is a tradeoff between efficiency and precision. By this way, the time complexity of *PR_prop* drops to $O(k^r)$, where constant $r \in \mathcal{N}$ is the radius of propagation. In the following part of this paper, pruning technique is always adopted by PageSim.

5.2. Complexity Analysis

The space complexity benefits from pruning technique too. Although the feature vector of a web page is designed to store PR scores of all web pages, the size of it is should be far less than n. Because on the huge Web, it is unlikely that a web page receives PR scores of all the pages, especially when the radius of propagation is "pruned". It is easy to conclude that the expectation of feature vector's size is also $O(k^r)$. As a result, the time complexity of *PS_calc* function is $O(k^r)$ too.

However, no matter what the complexity of *PS_calc* is, calculating PS scores of all n^2 page-pairs of the Web is a really tough task. Fortunately, based on the pruning assumption, pre-computing all PS scores can be avoided. As we know, a web page only stores $O(k^r)$ web pages' PR scores. On the other hand, a web pages's PR score can only be propagated to $O(k^r)$ pages. Therefore, the number of web pages which may contain common PR scores with a query page is $O(k^r) \cdot O(k^r) = O(k^{2r})$, which is also the time complexity of finding all of these pages. We can see that if $O(k^{2r})$ is small enough, the PageSim scores related to a query page can be computed *on-line*.

In conclusion, by adopting the pruning technique, the space complexity of PageSim is $O(Cn)$, the time complexity of propagating all of n web pages's PR scores is $O(Cn)$, and the time complexity of computing all of the PageSim scores related to a query page is $O(C^2)$, where $C = k^r$ is constant with respect to n.

5.3. Characteristics of PageSim

Based on the previous analysis, we deduce the inherent characteristics of PageSim, which enable it to be an applicable similarity measure for the Web.

Efficiency Apparently, the key factor of the complexities of PageSim is the propagation radius r, because large r may result in huge C which may dramatically increase the complexities of PageSim. Therefore, finding a smallest r while preserving the precision of PageSim is an important task. The experiments conducted in section 6 show that $r = 3$ is such an empirical propagation radius. Since averagely each web page has less than 10 links, accordingly $C < 10^3$. This indicates that our algorithm is efficient in both time and storage.

Scalability PageSim inherits parallelism property, because each web page propagates feature information *independently*. This property is very important, since PageSim can be implemented to utilize the computing power and storage capacity of tens to thousands of computers interconnected with a fast local network.

Stability The stability of PageSim is based on two aspects: the stability of PageRank and the "localism" of PageSim. First, in [14], the authors proved that the perturbed PR scores will not be far from the original so long as the perturbed web pages did not have high overall PR scores. This means that PageRank scores are fairly stable since web pages which have high PR scores are only a small part of the Web. Secondly, due to the pruning technique, web pages only propagate PR scores to their nearby neighbors, which means a small change of the Web only influences on the feature vectors of nearby web pages. Therefore we can conclude that it is *propagating stable PR scores locally* that makes PageSim stable.

Robustness First, PageSim is robust against text spamming since it is a pure link-based algorithm. Second, we illustrate the robustness of PageSim by showing that PageSim is resistant to *link farm*, which is a commonly used link spamming technique. A link farm is a network of web pages which are densely connected with each other [19]. It aims to boost the ranking of target web pages.

It is true that setting up sophisticated link structures within a link farm does not improve the *total PageRank of the link farm* [5], which is denoted by *PRLF*. As we know, the PageSim score between two web pages is less than the sum of *common* PR scores which originally propagated from *common* web pages (refer to equation (3)). Therefore, if a link farm links to two web pages (i.e., all the web pages in the link farm link to the two pages), its total effects on the PageSim score of these two pages is less than *PRLF*, which implies PageSim is robust against link farm. From the above analysis, we can see that by adopting the PR scores, PageSim indeed inherits a relatively strong ability of spamming resistance.

We list some other properties of PageSim below, which can be easily deduced from the definitions in section 4. For any web page u and v,

1. The PageSim scores are *symmetric*, i.e., $PS(u, v) = PS(v, u)$;
2. Each page is most similar to itself, i.e., $PS(u, u) = \max_{v \in V} PS(u, v)$;
3. $PS(u, v) \in [0, 1]$.

6. Preliminary Experimental Results

We have proposed an algorithm for measuring similarity between web pages. In this section, we report on some preliminary experimental results. The primary purpose is to show that PageSim scores indeed reflect degree of similarity between web pages. Experiments are also conducted to estimate the empirical value of propagation radius r of the algorithm and test the effect of the decay factor d on the result of PageSim.

A good evaluation of PageSim or any other similarity measure is difficult without performing extensive user studies or having a reliable ground truth for web page similarity. In this paper, we take a simple approach that uses the cosine TFIDF, a traditional text-based similarity function, as rough metrics of similarity. In spite of its simpleness, this approach does serve to illustrate the important aspects of PageSim empirically.

Given a web graph $G = (V, E)$, a similarity measure A produces a set of top T web pages most similar to page v (excluding v itself), which is denoted by $top_{A,T}(v)$. Let the number $sim_{A,T}(v)$ denotes the average cosine TFIDF similarity score to v of the $top_{A,T}(v)$. Thereby, we consider the average number of $sim_{A,T}(v)$ for all $v \in V$ as the quality of the top T web pages produced by algorithm A, which is denoted by $\Delta(A, T)$.

Cosine TFIDF Similarity The cosine TFIDF similarity score of two web pages u and v is just the cosine of angle between TFIDF vectors of the pages [8], which is defined by

$$sim(u, v) = \frac{\sum_{t \in u \cap v} W_{tu} \cdot W_{tv}}{\|u\| \cdot \|v\|}$$

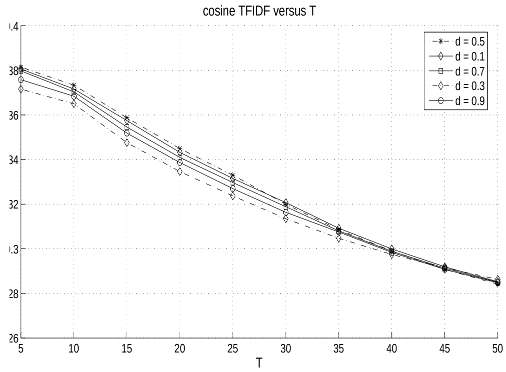

Figure 2. Quality of PageSim for different d

where W_{tu} and W_{tu} are TFIDF weights of term t for web page u and v respectively. $\|v\|$ denotes the length of page v, which is defined by $\|v\| = \sqrt{\sum_{t \in v} W_{tv}^2}$.

Data Set The data set used in the experiments is a set of web pages crawled from *http://www.cse.cuhk.edu.hk*, the web site of CSE department of CUHK. The web graph contains more than 20,000 web pages with about 180,000 hyperlinks linking them together.

6.1. Experiments on the Decay Factor

We first check the effect of the constant d in equation (2) on the result of PageSim. We mentioned that d is a "decay factor". The intuition behind d is simple and natural: the feature information propagated to distant pages should decrease during the propagation.

In the following experiment on d, we set $r = n - 1$. Figure 2 plots the curves for different values of d, with the x-axis representing the value of T and y-axis representing the value of score $\Delta(PageSim, T)$. The downward curves show a decrease in score as T increases. It shows that the effect of d is not significant since the curves are very close. Relatively, the curve corresponding to $d = 0.5$ is the best, so we set $d = 0.5$ in the following experiments of this paper.

6.2. Experiments on the Propagation Radius

In section 5.2, we proposed the pruning technique to reduce the complexities of PageSim. Certainly, the quality of the pruning approximation must be verified experimentally. Therefore, we conduct the following experiments to reveal the effects of the propagation radius r on the results of PageSim and get an empirical radius.

Figure 3 plots the curves for different values of r. We can see that the curve of $r = 3$ is very close to the "actural" curve of PageSim. This verified our assumption that the PageRank scores propagated to the web pages more than 3 hops long is small enough to be omitted. Empirically, we can choose $r = 3$ to be the propagation radius in practice to improve the efficiency of PageSim. Since averagely each web page has 7 to 10 links, k is less than 10. Accordingly, we have $C < 1,000$.

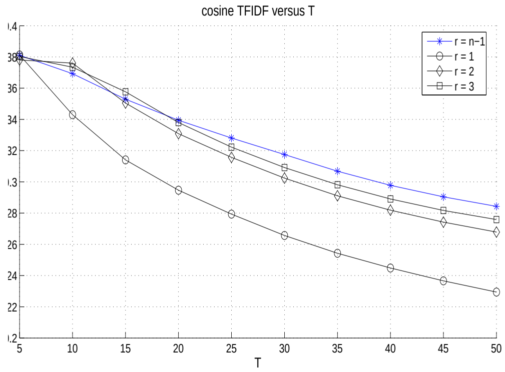

Figure 3. Quality of PageSim for different r

6.3. PageSim and SimRank

In this part, we compare PageSim with the SimRank which is a well-defined similarity measure. SimRank is a fixed point of the recursive definition: *two objects are similar if they are referenced by similar objects*. Numerically, this is specified by defining $simrank(u, u) = 1$ and

$$simrank(u, v) = \gamma \cdot \frac{\sum_{a \in I(u)} \sum_{b \in I(v)} simrank(a, b)}{|I(u)||I(v)|} \tag{5}$$

for $u \neq v$ and $\gamma \in (0, 1)$, where $I(x)$ denotes the set of inlink vertices of x. 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 $lim_{k \to \infty} simrank_k(u, v)$, the proof of convergence can be found in [7].

Figure 4 plots the curves of PageSim and SimRank. For the curve of PageSim, we set $r = 3$ and $d = 0.5$. Across all T, the average improvement of PageSim over SimRank under the cosine TFIDF measure is about 8%.

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 web pages. There are numbers of future directions. Foremost, more extensive experiments are needed to evaluate the performance of PageSim. Secondly, we believe that

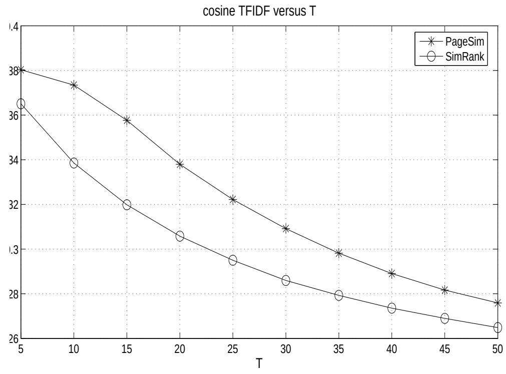

Figure 4. Quality of PageSim and SimRank

a practical similarity measure must be hybrid, so integrating PageSim with other (text-based) similarity measure is another direction of future work.

8. Acknowledgment

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

[1] http://www.google.com.

[2] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. Searching the Web. *ACM Trans. Inter. Tech.*, 1(1):2–43, 2001.

[3] B. Brewington, G. Cybenko, R. Stata, K. Bharat, and F. Maghoul. How dynamic is the Web? In *WWW '00: Proceedings of the 9th Conference on World Wide Web*, Amsterdam, Netherlands, May 2000. ACM Press.

[4] G. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the 6th ACM SIGKDD*, pages 150–160, Boston, MA, USA, 2000.

[5] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *the First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[6] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[7] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *KDD '02: Proceedings of the 8th ACM SIGKDD*, pages 538–543, New York, NY, USA, 2002. ACM Press.

[8] T. Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 143–151, San Francisco, CA, USA, 1997.

[9] M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(10–25), 1963.

[10] S. Lawrence and C. L. Giles. Accessibility of information on the Web. *Intelligence*, 11(1):32–39, 2000.

[11] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *12th International Conference on Information and Knowledge Management*, pages 556–559. ACM, November 2003.

[12] W. Lu, J. Janssen, E. Milios, and N. 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.

[13] M. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter*, 38(2):321–330, 2004.

[14] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–266, NY, USA, 2001. ACM Press.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[16] K. Risvik and R. Michelsen. Search engines and web dynamics. *Computer Networks*, 39(3):289–302, 2002.

[17] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

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

[19] B. Wu and B. D. Davison. Identifying link farm spam pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 820–829, NY, USA, 2005. ACM Press.