13

Link Spam Target Detection Using Page Farms

BIN ZHOU and JIAN PEI
Simon Fraser University

Currently, most popular Web search engines adopt some link-based ranking methods such as PageRank. Driven by the huge potential benefit of improving rankings of Web pages, many tricks have been attempted to boost page rankings. The most common way, which is known as link spam, is to make up some artificially designed link structures. Detecting link spam effectively is a big challenge. In this article, we develop novel and effective detection methods for link spam target pages using page farms. The essential idea is intuitive: whether a page is the beneficiary of link spam is reflected by how it collects its PageRank score. Technically, how a target page collects its PageRank score is modeled by a page farm, which consists of pages contributing a major portion of the PageRank score of the target page. We propose two spamicity measures based on page farms. They can be used as an effective measure to check whether the pages are link spam target pages. An empirical study using a newly available real dataset strongly suggests that our method is effective. It outperforms the state-of-the-art methods like SpamRank and SpamMass in both precision and recall.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms: Algorithms

Additional Key Words and Phrases: PageRank, Page Farm, Link Spam

ACM Reference Format:

Zhou, B. and Pei, J. 2009. Link spam target detection using page farms. ACM Trans. Knowl. Discov. Data. 3, 3, Article 13 (July 2009), 38 pages.
DOI = 10.1145/1552303.1552306 http://doi.acm.org/10.1145/1552303.1552306

1. INTRODUCTION

Ranking Web pages is an essential task in Web search and mining. Many studies have been dedicated to effective ranking methods such as HITS

This research is supported in part by an NSERC Discovery Grant, an NSERC Discovery Accelerator Supplements Grant, and a Microsoft Research Grant. All opinions, findings, conclusions and recommendations in this article are those of the authors and do not necessarily reflect the views of the funding agencies.
Authors' address: School of Computing Science, Simon Fraser University, Canada; email: {bzhou, jpei}@cs.sfu.ca.

DOI 10.1145/1552303.1552306 http://doi.acm.org/10.1145/1552303.1552306

[Kleinberg 1999] and PageRank [Page et al. 1998]. Most popular Web search engines currently adopt some link-based ranking methods such as PageRank. Driven by the huge potential benefit of promoting rankings of Web pages, many tricks have been attempted to boost page rankings. The most common way, which is known as link spam [Bianchini et al. 2005; Gyöngyi and Garcia-Molina 2005b; Henzinger et al. 2003; Langville and Meyer 2004], is to make up some artificially designed link structures. Generally, a Web page may benefit from link spam activities, or it may support link spam activities. In this paper, we are focusing on finding those pages which are benefiting from link spam, that is, the targets of link spam campaigns. We call such a page a *link spam target page*. We use the term *link spam* to refer to the activity/phenomena such that spammers try to mislead search engines to boost the rankings of the target pages.

Detecting link spam effectively is a big challenge. Several link spam detection methods were developed in some previous studies. Please see Section 6 for a brief review. However, two important and interesting questions largely remain open.

First, owners of Web pages often want to promote their Web pages and boost the rankings by attracting links from other Web sites. The difference between a popular page and a link spam target page depends on whether the links from other pages are justifiable. Thus, it is often relative and subtle whether a Web page is a link spam target page. To this extent, link spam detection is different from the traditional classification/supervised learning problem. Instead of simply classifying a page as a link spam target page or not, alternatively, can we measure the "degree" that a page is a link spam target page? What features can be used to measure the degree well?

Moreover, although some methods have been proposed to detect link spam, there exists no method to detect how link spam is attempted by a link spam target page, that is, how hyperlinks are used to connect the supporting pages and the target page so that the PageRank score of the target page is boosted. How link spam is attempted is important for understanding link spam better and improving ranking methods in Web search engines.

To fully understand link-based ranking, one essential question largely remains open. For a Web page p, what other pages are the major contributors to the ranking score of p, and how is the contribution made? Understanding the general relations of Web pages and their environments is important, with quite a few interesting applications such as link spam detection.

In this article, we study the problem of link spam target page detection and propose novel and effective link spam detection methods using page farms. The essential idea is intuitive: whether a page is a link spam target page is reflected by how it collects its PageRank score. Technically, for a target page p, we model how p collects its PageRank score by the page farm of p which consists of pages contributing to a major portion of the PageRank score of p. By analyzing the utility and characteristics of the page farm of p in boosting the PageRank score of p, we derive the spamicity measure of p, which reflects the "degree" of link spam and can be used as an effective measure of link spam.

Gyöngyi and Garcia-Molina [2005a] modeled the Web pages and the hyperlinks supporting a spam target page as the link spam farm of the spam target page. The ideas of page farms and link spam farms share some similarity. However, there are some critical differences between link spam farms and page farms. The link spam farm model in Gyöngyi and Garcia-Molina [2005a] is a conceptual notion. To the best of our knowledge, there are no previous studies on what exactly a link spam farm looks like in practice and how to extract those link spam farms. In our page farm model, every page has its own page farm which is well defined and contains those most important pages that contribute to a major part of the PageRank score of the target page. To this extent, our page farm notion is more general than the link spam farm notion in scope, and more detailed in technical definition.

An important application of page farms is detecting link spam target pages. The page farm model and the related methods presented in this article have some advantages. First, page farms not only can capture link spam, but also can tell how link spam is attempted. Second, our methods using page farms to detect link spam can capture disguised link spam. Some existing methods, as briefly reviewed in Section 6, can identify optimal link spam farms. However, such perfect link spam farms may not be commonly used in practice since they can be detected easily by a search engine. To disguise, a spammer may modify the optimal link spam farm but still keep the target pages of high PageRank scores. By analyzing page farms, we can still capture disguised link spam since the disguised link spam still has a page farm efficiently boosting the PageRank score of the target page. Third, as will be shown in the article, extracting the page farm of a Web page is efficient using a simple yet effective local greedy search method. We do not need the whole Web graph or a costly training procedure. Thus, page farm based detection methods can be applied on the fly on any target pages of interest.

We make the following contributions.

—We propose page farm analysis for link spam detection, as discussed in Section 2 and Section 3. A page farm is a (minimal) set of pages contributing to (a major portion of) the PageRank score of a target page. We propose the notions of θ-page farms and (θ, k)-page farms, where θ in $[0, 1]$ is a contribution threshold and k is a distance threshold. We study the computational complexity of finding page farms, and show that it is NP-hard. Then we develop a feasible greedy method in practice to extract approximate page farms for a set of pages. The method potentially can be extended to extracting page farms for all pages in the whole Web graph.

—We investigate link spam detection using page farms, as shown in Section 4. In the utility-based method, we measure the utility of a page farm, that is, the "perfectness" of a page farm in obtaining the maximum PageRank. Among those pages that try to achieve PageRank scores as much as possible, the majority of them can be classified into link spam pages. Thus, the utility can be used as a measure of the likelihood of link spam. In the characteristics-based method, we analyze the characteristics of a page farm, that is, the statistics of a page farm such as how pages and hyperlinks are organized in the farm, and use the statistics as the indicator of the likelihood of link spam. Using those measures we can detect link spam target pages. Different from many previous methods, our methods are not based on classification or supervised training.

—We evaluate our link spam detection methods using a newly available large real dataset, as shown in Section 5. The dataset [Castillo et al. 2006] was released by Yahoo! Research Barcelona, which is the result of the effort of an international team of volunteers. As far as we know, this dataset is the first publicly available benchmark for Web spam detection. The experimental results show that our methods are effective in detecting spam pages and outperform SpamRank and SpamMass which are the state-of-the-art methods that assign to pages spamicity scores and do not need supervised training.

The rest of the article is organized as follows. In Section 2, we present the page farm model. In Section 3, we discuss page farm extraction. Section 4 investigates link spam detection using page farms. We report an empirical evaluation in Section 5, and review the related work in Section 6. The article is concluded in Section 7.

2. PAGE FARMS

Web pages and hyperlinks can be modeled as a directed *Web graph* $G = (V, E)$, where V is the set of Web pages, and E is the set of hyperlinks. A link from page p to page q is denoted by the edge $p \rightarrow q$. An edge $p \rightarrow q$ can also be written as a tuple (p, q). A page p may have multiple hyperlinks pointing to page q, however, in the Web graph, only one edge $p \rightarrow q$ is formed. Hereafter, by default our discussion is about a directed Web graph $G = (V, E)$.

PageRank [Page et al. 1998] measures the importance of a page p by considering how collectively other Web pages point to p directly or indirectly. Formally, for a Web page p, the PageRank score is defined as

$$PR(p, G) = d \sum_{p_i \in M(p)} \frac{PR(p_i, G)}{OutDeg(p_i)} + \frac{1-d}{N}, \tag{1}$$

where $M(p) = \{q | q \rightarrow p \in E\}$ is the set of pages having a hyperlink pointing to p, $OutDeg(p_i)$ is the outdegree of p_i (that is, the number of hyperlinks from p_i pointing to some pages other than p_i), d is a *damping factor* which models a decay in relevance/trust over distance on the Web, and $N = |V|$ is the total number of pages in the Web graph. $1 - d$ is traditionally referred to as the random jump probability. The second additive term on the right hand side of the equation, $\frac{1-d}{N}$, corresponds to a minimal amount of PageRank score that every page gets by default.

To calculate the PageRank scores for all Web pages in a graph, one can assign a random PageRank score value to each node in the graph, and then apply Equation (1) iteratively until the PageRank scores in the graph converge.

For a Web page p, can we analyze which other pages contribute to the PageRank score of p? A straightforward way to answer the above question is to extract *all* Web pages that contribute to the PageRank score of the target page p. This idea leads to the notion of page farms.

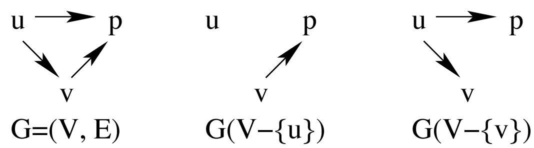


Fig. 1. Voiding pages and induced subgraphs. $G(V-\{u\})$ and $G(V-\{v\})$ are the induced subgraphs of G on $V-\{u\}$ and $V-\{v\}$, respectively, as specified in Definition 2.2.

Definition 2.1 (*Page Farm*). For a page p, the page farm of p is the set of pages on which the PageRank score of p depends. Page p is called the target page.

Please note that when loops are present such that p contributes to its PageRank, p itself can also be in the page farm of p.

According to Equation (1), the PageRank score of p directly depends on the PageRank scores of pages having hyperlinks pointing to p. The dependency is transitive. Therefore, a page q is in the page farm of p if and only if there exists a directed path from q to p in the Web graph.

As indicated in some previous studies [Albert et al. 1999; Broder et al. 2000], the major part of the Web is strongly connected. Albert et al. [1999] indicated that the average distance of the Web is 19. Moreover, Adamic et al. [1999] claimed that the Web also has a small world topology. In other words, it is highly possible to get from any page to another in a small number of clicks. A strongly connected component of over 56 million pages (within a crawl of 203 million pages in total) is reported in Broder et al. [2000]. Therefore, the page farm of a Web page can be very large. It is difficult to analyze page farms of a large number of pages. On the other hand, in many cases one may be interested in only the major contributors to the PageRank score of the target page.

Can we capture the set of major contributors to a large portion of the PageRank score of a target page?

According to Equation (1), PageRank contributions are only made by the out-links. Thus, a vertex in the Web graph is *voided* for PageRank score calculation if all edges leaving the vertex are removed. Please note that we cannot simply remove the vertex. Consider Graph G in Figure 1. Suppose we want to void page v in the graph for PageRank calculation. Removing v from the graph also reduces the outdegree of u, and thus changes the PageRank contribution from u to p. Moreover, simply removing v alters the random jump probability of each page in Figure 1. The effect is undesirable. Instead, we should retain v but remove the out-link $v \rightarrow p$.

Definition 2.2 (*Induced Subgraph*). For a set of vertices U, the induced subgraph[1] of U (with respect to PageRank score calculation) is given by $G(U) = (V, E')$, where $E' = \{p \rightarrow q | (p \rightarrow q \in E) \wedge (p \in U)\}$. In other words, in $G(U)$, we void all vertices that are not in U. Figure 1 shows two examples.

[1]Please note that the definition of the induced subgraph here is different from the conventional definitions of edge/vertex-induced subgraphs in graph theory.

To evaluate the contribution of a set of pages U to the PageRank score of a page p, we can calculate the PageRank score of p in the induced subgraph of U. Then, the *PageRank contribution* is

$$Cont(U, p) = \frac{PR(p, G(U))}{PR(p, G)} \times 100\%,$$

where $PR(p, G(U))$ and $PR(p, G)$ represent the PageRank scores of p in $G(U)$ and G, respectively.

It is crucial to handle the dangling nodes properly in PageRank calculation. The pages of the Web can be classified as either dangling nodes or nondangling nodes. A *dangling node* is a Web page that contains no outlinks. All other pages, having at least one outlink, are called *nondangling nodes*. Dangling nodes affect the PageRank calculation because "it is not clear where their weight should be distributed" [Page et al. 1998]. There are several ways to treat the dangling nodes. The goal of the fixes for dangling nodes is to ensure that the sum of PageRank scores of all nodes in a graph is equal to 1. Two well-known and equivalent approaches are as follows [Langville and Meyer 2004]. We can link every dangling node to all other nodes in a graph. Alternatively, we can link every dangling node to a virtual node that links to all other nodes in a graph.

Here, our goal is to examine the contribution of a (small) subset of vertices to the PageRank of a target vertex. This is essentially different from the goal of fixing dangling nodes in the traditional PageRank model. Technically, when some vertices are voided, they become dangling nodes in the induced subgraphs. Conceptually, the PageRank potential going to those vertices should be distributed evenly to all vertices in the graph. When the graph is very large and the page farm is comparatively very small, only a very small part of the PageRank potential going to those dangling vertices can go back to the target vertex, which thus can be ignored in the analysis. Therefore, in our model, we do not link those dangling vertices to any other vertices.

If we adopt the traditional PageRank model to fix the dangling vertices in an induced subgraph, by linking those dangling vertices to the other vertices in the induced subgraph in one way or another, there still exist some link pathes from the dangling vertices to the target vertex, thus the contribution of those dangling vertices as well as their close neighbors to the target vertex is over counted. Therefore, we do not adopt the fixes of dangling vertices in the traditional PageRank model.

PageRank contribution has the following property, which follows from Corollary 3.3, as discussed later in Section 3.

COROLLARY 2.3 (MONOTONIC CONTRIBUTIONS). *Let p be a page and U, W be two sets of pages such that $U \subset W$. Then,* $0 \leq Cont(U, p) \leq Cont(W, p) \leq 1$.

We can use the smallest subset of Web pages that contribute to at least a θ portion of the PageRank score of a target page p as its θ-(page) farm.

Definition 2.4 (θ*-farm*). Let θ be a parameter such that $0 \leq \theta \leq 1$. A set of pages U is a θ*-farm* of page p if $Cont(U, p) \geq \theta$ and $|U|$ is minimized.

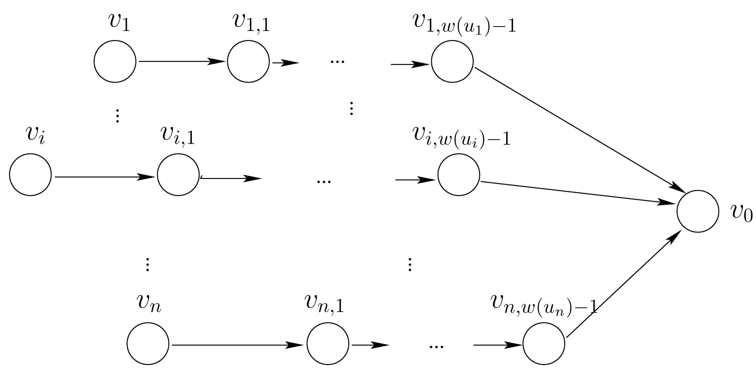


Fig. 2. Reducing the knapsack problem to a θ-farm extraction problem.

However, finding a θ-farm of a page is computationally costly on large networks, as indicated by the following result.

THEOREM 2.5 (θ-FARM COMPLEXITY). *The following decision problem is NP-hard: for a Web page p, a parameter θ, and a positive integer n, determine whether there exists a θ-farm of p which has no more than n pages.*

PROOF. The theorem can be proved by reducing the knapsack problem, which is NP-complete [Karp 1972], to the θ-farm extraction problem.

We are given a set U of n items u_i $(1 \leq i \leq n)$ with value $val(u_i)$ and weight $w(u_i)$, where the values and the weights are positive integers. We are asking whether there exists a subset of items $S \subseteq U$ such that the total value of the subset is at least K, that is, $\sum_{u \in S} val(u) \geq K$, and the total weight of the subset is at most W, that is, $\sum_{u \in S} w(u) \leq W$, where K and W are given positive integers.

To reduce the problem, we construct a directed graph $H = (V, E)$. The vertices and the edges are created in three steps.

(1) First, vertex $v_0 \in V$ is created as a "knapsack."
(2) Second, for each item u_i, we create a vertex v_i.
(3) Last, for each vertex $v_i \in V$ $(1 \leq i \leq n)$ created in the second step, we construct a directed path from v_i to v_0 of length $w(u_i)$: $(w(u_i)-1)$ new vertices, denoted by $v_{i,1}, \ldots, v_{i,w(u_i)-1}$, are inserted as a path $v_i \to v_{i,1} \to v_{i,w(v_i)-1} \to v_0$.

Figure 2 illustrates the construction of the graph H. As the result, in H, the set of vertices $|V| = 1 + n + \sum_{j=1}^{n}(w(u_i) - 1)$ and $|E| = \sum_{j=1}^{n} w(u_i)$.

We compute the PageRank scores of the vertices by assigning the initial score values to the vertices in graph H as follows: for each vertex v_i, where $1 \leq i \leq n$, $PR(v_i) = val(u_i)$. All the other vertices (that is, v_0 and $v_{i,j}$'s) are assigned an initial score 0. We set $d = 1$ in Equation (1) and compute the PageRank scores of the nodes in the graph.

Under such an initial score assignment, the PageRank score of v_0 has the following properties. First, vertices $v_{i,1}, \ldots, v_{i,w(u_i)-1}$ contribute to v_0's PageRank score if and only if the complete path $v_i \rightarrow v_{i,1} \rightarrow v_{i,w(v_i)-1} \rightarrow v_0$ is retained in the induced subgraph. In other words, v_0 can obtain some positive contribution from any subset of the nodes in this path only if the whole path is included in the farm. If only some nodes in the path are included in the farm, the farm is not minimal since removing those nodes reduces the size of the farm but the PageRank score of v_0 remains.

Second, for a graph $H' \subseteq H$ which contains only a path $v_i \rightarrow v_{i,1} \rightarrow v_{i,w(v_i)-1} \rightarrow v_0$, the converged PageRank score of v_0 in H' is $val(u_i)$.

Last, in graph $H' \subseteq H$ which contains directed paths from $v_{j_1}, \ldots, v_{j_l}$ to v_0 ($1 \leq j_1, \ldots, j_l \leq n$), the converged PageRank score of v_0 is $\sum_{i=1}^{l} val(u_{j_i})$. Moreover, $PR(v_0, G) = \sum_{i=1}^{n} val(u_i)$.

Therefore, we obtain an affirmative answer to the knapsack problem (that is, there is a set of items whose sum of values is at least K and whose sum of weights is at most W) if and only if, in the transformed graph G, there is a $\frac{K}{PR(v_0,G)}$-farm of v_0 of size at most W.

Please note that we do not need to explicitly unfold those paths from v_i to v_0 in the real graph for PageRank score calculation and page farm computation. Vertices v_i's are the representatives of the paths. Therefore, the transformation is of polynomial complexity. □

Searching many pages on the Web can be costly. Heuristically, the near neighbors of a Web page often have strong contributions to the importance of the page, since the contribution from one page to another decays dramatically as the distance from the contributor to the beneficiary increases. The decay can be captured by the concepts of page contribution (Definition 3.1) and path contribution (Definition 3.2). Therefore, we propose the notion of (θ, k)-farm.

In a directed graph G, let p, q be two nodes. The *distance* from p to q, denoted by $dist(p, q)$, is the length (in number of edges) of the shortest directed path from p to q. If there is no directed path from p to q, then $dist(p, q) = \infty$.

Definition 2.6 (*(θ, k)-Farm*). Let $G = (V, E)$ be a directed graph. Let θ and k be two parameters such that $0 \leq \theta \leq 1$ and $k > 0$. k is called the distance threshold. A subset of vertices $U \subseteq V$ is a (θ, k)-farm of a page p if $Cont(U, p) \geq \theta$, $dist(u, p) \leq k$ for each vertex $u \in U$, and $|U|$ is minimized.

Unfortunately, we notice that finding the exact (θ, k)-farms is also computationally expensive on large networks, as shown in Corollary 2.7.

COROLLARY 2.7 ((θ, k)-FARM COMPLEXITY). *The following decision problem is NP-hard: for a Web page p, a parameter θ, a distance threshold k, and a positive integer n, determine whether there exists a (θ, k)-farm of page p having no more than n pages.*

PROOF. (θ, k)-farm is a special case of θ-farm, since we can set k to the eccentricity of the graph. Thus, finding (θ, k)-farm is also NP-hard. □

Thus, we have to develop efficient heuristic methods, which is the task of the next section. In the rest of the paper, for the sake of brevity, when a page farm is mentioned we refer to its (θ, k)-farm.

Since PageRank is the most popular link based ranking algorithm, in this paper, we focus on PageRank-based page farms and their extraction.

3. PAGE FARM EXTRACTION

In this section, we first give a simple greedy method to extract page farms, and analyze why the simple greedy method is inefficient. Then, we propose a method feasible in practice to extract approximate page farms.

3.1 A Simple Greedy Method

Clearly, if we can measure the contribution from any single page v towards the PageRank score of the target page p, then we can greedily search for pages of big contributions and add them into the page farm of p.

Definition 3.1 (*Page Contribution*). For a target page $p \in V$, the page contribution of page $v \in V$ to the PageRank score of p is

$$PCont(v, p) = \begin{cases} PR(p, G) - PR(p, G(V - \{v\})) & (v \neq p) \\ \dfrac{1-d}{N} & (v = p) \end{cases}$$

where d is the damping factor, and N is the total number of pages in the Web graph.

Definition 3.1 is based on an intuitive observation, and is reasonable and easy to understand. If $v = p$, according to the original PageRank formula in Equation (1), $\frac{1-d}{N}$ corresponds to a minimal amount of PageRank score that every page gets by default. Thus, we define $PCont(v, p) = \frac{1-d}{N}$. If $v \neq p$, intuitively, the PageRank contribution from v to p is the decrease of the PageRank score of page p after we void page v. Thus, we define $PCont(v, p) = PR(p, G) - PR(p, G(V - \{v\})$.

While PageRank contribution defined in Section 2 is used to measure the ratio of contributions from a set of pages to a target page, page contribution is used to measure the contribution from a specific page to a target page. To quantified the contributions from a single page, we define page contribution using differences rather than fractions.

Example 1 (*Page Contribution*). Consider the simple Web graph G in Figure 1. The induced subgraphs $G(V - \{u\})$ and $G(V - \{v\})$ are also shown in the figure. Please note that when a node is voided from the graph, the out-degrees of its neighbors remain intact in the PageRank formula.

Let us consider page p as the target page, and calculate the page contributions of the other pages to the PageRank of p. In this example, $N = 3$. According to Equation (1), the PageRank score of p in G is given by

$$PR(p, G) = -\frac{1}{6}d^3 - \frac{1}{3}d^2 + \frac{1}{6}d + \frac{1}{3}.$$

Moreover, the PageRank score of p in $G(V - \{u\})$ is

$$PR(p, G(V - \{u\})) = -\frac{1}{3}d^2 + \frac{1}{3},$$

and the PageRank score of p in $G(V - \{v\})$ is

$$PR(p, G(V - \{v\})) = -\frac{1}{6}d^2 - \frac{1}{6}d + \frac{1}{3}.$$

Thus, the page contributions $PCont(u, p)$ and $PCont(v, p)$ are calculated as

$$PCont(u, p) = PR(p, G) - PR(p, G(V - \{u\})) = -\frac{1}{6}d^3 + \frac{1}{6}d,$$

and

$$PCont(v, p) = PR(p, G) - PR(p, G(V - \{v\})) = -\frac{1}{6}d^3 - \frac{1}{6}d^2 + \frac{1}{3}d.$$

Using the page contributions, we can greedily search a set of pages that contribute to a θ portion of the PageRank score of a target page p. That is, we calculate the page contribution of every page with distance to p at most k to the PageRank score of p, and sort the pages in the contribution descending order. Suppose the list is $u_1, u_2, \ldots$. Then, we select the top-l pages $u_1, \ldots, u_l$ as an approximation of the θ-farm of p such that $\frac{PR(p,G(V-\{u_1,\ldots,u_l\}))}{PR(p,G)} \geq \theta$ and l is minimized.

This *naïve greedy search* method is unfortunately inefficient for large Web graphs. First, it assumes that the whole Web graph is available, which may not be true for many situations for search engines. For example, the Web is dynamic. It is not easy to maintain a Web graph which is up to date. Second, in order to extract the page farm for a target page p, we have to compute the PageRank of p in the induced subgraph $G(V - \{q\})$ for every page q other than p. In the worst case, to extract the (θ, k)-farm of page p, if there are m pages q such that the distance from q to p is no more than k, then we need to compute the PageRank of p in m induced graphs. The computation is very costly since the PageRank calculation is an iterative procedure and often involves a huge number of Web pages and hyperlinks.

3.2 Path Contributions

Computing the contribution page by page is costly. Can we reduce this cost effectively? Our idea is to compute the contribution path by path.

Definition 3.2 (*Path Contribution*). Consider Web graph $G = (V, E)$ and target page $p \in V$. Let $P = v_0 \to v_1 \to \ldots \to v_n \to p$ be a (directed) path from v_0 to p in the graph. The *path contribution* to the PageRank of p from P is defined as

$$LCont(P, p) = \frac{1}{N}d^{n+1}(1 - d)\prod_{i=0}^{n}\frac{1}{OutDeg(v_i)},$$

where $OutDeg(v_i)$ is the outdegree of page v_i (the one from the original graph, since voiding a node in the graph does not change the out-degrees of its neighbors), and N is the total number of pages in the Web graph.

The PageRank score of p can be calculated using the path contributions [Brinkmeier 2006].

$$PR(p, G) = \frac{1-d}{N} + \sum_{v \in W(p)} \left(\sum_{P \in DP(v,p)} LCont(P, p) \right), \tag{2}$$

where $W(p) = \{v | \text{there is a directed path from } v \text{ to } p\}$, $DP(v, p) = \{\text{directed path } P \text{ from } v \text{ to } p\}$, and N is the total number of Web pages in the Web graph.

Moreover, page contributions can also be calculated using path contributions. Applying Equation (2) to Definition 3.1, we have the following result.

COROLLARY 3.3 (PAGE AND PATH CONTRIBUTIONS). *For vertices p and q in Web graph $G = (V, E)$, if the indegree of q is 0, that is, $InDeg(q) = 0$, then*

$$PCont(q, p) = \sum_{\text{path } P \text{ from } q \text{ to } p} LCont(P, p).$$

If $InDeg(q) > 0$, then

$$\begin{aligned} PCont(q, p) = & \sum_{\text{path } P_1 \text{ from } q \text{ to } p} LCont(P_1, p) \quad + \\ & \sum_{v \in W_q(p)} \sum_{\text{path } P_2 \text{ from } v \text{ to } p \text{ through } q} LCont(P_2, p), \end{aligned}$$

where $W_q(p) = \{v | \text{there is a directed path from } v \text{ to } p \text{ through } q\}$.

PROOF. According to Definition 3.1, for vertices p and q ($q \neq p$) in Web graph $G = (V, E)$,

$$PCont(q, p) = PR(p, G) - PR(p, G(V - \{q\})). \tag{3}$$

For simplicity, in the later analysis, we use $P \in G(V)$ to represent an arbitrary link path in $G(V)$. We apply Equation (2) to Equation (3), and have the following:

$$\begin{aligned} & PCont(q, p) \\ = & \left(\frac{1-d}{N} + \sum_{P \in G(V)} LCont(P, p) \right) - \left(\frac{1-d}{N} + \sum_{P' \in G(V - \{q\})} LCont(P', p) \right) \\ = & \sum_{P \in G(V)} LCont(P, p) - \sum_{P' \in G(V - \{q\})} LCont(P', p) \end{aligned} \tag{4}$$

Since the induced graph $G(V - \{q\})$ is generated by removing the out-links of q, if $InDeg(q) = 0$, the differences between the two sets of link paths $\mathcal{P}$ and

$\mathcal{P}'$ are those link paths from q to p. If $InDeg(q) > 0$, the differences between $\mathcal{P}$ and $\mathcal{P}'$ are those link paths from q to p, and those link paths to p through q. Thus, based on Equation (4), we have Corollary 3.3. □

Please note that some paths in Corollary 3.3 can be circular. All the paths, including those generated by loops, have to be considered in the calculation.

The monotonic contribution property of PageRank contributions (Corollary 2.3) can be derived from Corollary 3.3.

PROOF OF COROLLARY 2.3. In order to void a vertex in the graph, we need to remove all of its outlinks but keep the vertex in the graph. Note that those voided vertices become dangling vertices in PageRank terminology. However, in our PageRank contribution model, we do not need to make any specific modifications to remove dangling node.

According to Definition 3.2, a path contribution is non-negative. When some vertices are voided for PageRank calculation, some paths (pointing to the target vertex) are destroyed. Thus, the PageRank score of p in the induced subgraph cannot be larger than that in the original graph, and the PageRank contribution is a number between 0 and 1. □

Example 2 (*Path Contribution*). Consider the Web graph in Figure 1 again. There are three paths to target page p: $P_1 : u \to p$, $P_2 : u \to v \to p$, and $P_3 : v \to p$. The path contributions can be calculated as

$$\begin{aligned} LCont(P_1, p) &= -\frac{1}{6}d^2 + \frac{1}{6}d, \\ LCont(P_2, p) &= -\frac{1}{6}d^3 + \frac{1}{6}d^2, \end{aligned}$$

and

$$LCont(P_3, p) = -\frac{1}{3}d^2 + \frac{1}{3}d.$$

Using Corollary 3.3, we have

$$PCont(u, p) = LCont(P_1, p) + Lcont(P_2, p) = -\frac{1}{6}d^3 + \frac{1}{6}d$$

and

$$PCont(v, p) = LCont(P_3, p) + LCont(P_2, p) = -\frac{1}{6}d^3 - \frac{1}{6}d^2 + d.$$

Moreover, by Equation (2), we have

$$\begin{aligned} PR(p, G) &= \frac{1-d}{3} + LCont(P_1, p) + LCont(P_2, p) + LCont(P_3, p) \\ &= -\frac{1}{6}d^3 - \frac{1}{3}d^2 + \frac{1}{6}d + \frac{1}{3}. \end{aligned}$$

The results are consistent with those in Example 1.

Compared to page contributions, path contributions are cheaper to compute and approximate. We can derive them directly from the graph structure

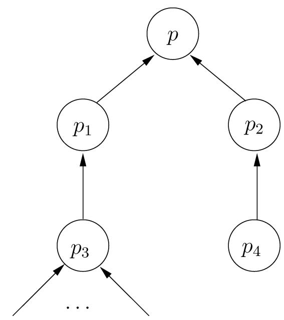


Fig. 3. An example showing the different results using two measures of PageRank contribution.

using the outdegrees of pages. We do not even need the PageRank scores of pages. Moreover, path contributions can be calculated in an incremental way. There may be an infinite number of paths due to loops. However, according to Definition 3.2, when the length of the path increases, the path contribution decreases dramatically. In practice, we don't need to consider all the paths. Furthermore, recall that computing a page contribution directly using Definition 3.1 has to iteratively compute the PageRank score of the target page in an induced subgraph until the scores in the subgraph converge.

Interestingly, simultaneous to our study on page contribution and path contribution, Gyöngyi et al. [2006] proposed a measure on PageRank contribution from Web pages and link paths. In their definition, the contribution from page q to page p is given by $\sum_{P \in DP(q,p)} LCont(P, p)$. The critical difference is that their definition does not consider the transitive contribution from links pointing to q (that is, the second item in Corollary 3.3).

To illustrate the difference, consider the directed graph in Figure 3. Suppose there are many links pointing to p_3, which in turn makes page p_1 contribute much more than p_2 to the PageRank of p. As a result, using our PageRank contribution model, pages p_1 and p_3 will be included in the farm. However, in the model in Gyöngyi et al. [2006], p_1 and p_2 have the same PageRank contribution, and p_2's contribution is much larger than p_3. This is due to the reason that the path from p_3 to p is larger than that from either p_1 or p_2 to p. Consequently, the page farm may include either p_1 and p_2 into the farm but not p_3. The purposes of their model and ours are different. Their model measures individual impact exclusively. Ours captures the most important PageRank contributors in the graph.

We also notice that, simultaneous to our study on path contribution, a similar idea called branching contribution is presented in Baeza-Yates et al. [2006]. The difference is that Baeza-Yates et al. [2006] only consider PageRank contributions over link paths, but does not model the PageRank contributions from pages explicitly.

3.3 Extracting (θ, k)-farms

Using path contributions, we propose a local greedy search algorithm in Figure 4. It takes the immediate neighbors of p (that is, those pages having

Input: a Web graph $G = (V, E)$, target page $p \in V$, a damping factor d, parameters θ and k;
Output: an approximate (θ, k)-farm of page p;
Method:

initialize $Farm = \emptyset$;
let $S = \{v | v \rightarrow p \in E\}$;
WHILE $(PageRankContribution(Farm, p) < \theta)$ DO {
  IF $S = \emptyset$ THEN RETURN $\emptyset$; // no farm is found
  $q = \arg\max_{q \in S}\{PCont(q, p)\}$;
  $Farm = Farm \cup \{q\}$;
  $S = (S - \{q\}) \cup \{q' | (q' \rightarrow q \in E) \wedge (dist(q', p) \leq k) \wedge (q' \notin Farm)\}$;
}
RETURN $Farm$;

Function $PageRankContribution(Farm, p)$
// Compute the PageRank contribution from pages in $Farm$
compute $PR(p, G(Farm \cup \{p\}))$, the PageRank of p in $G(Farm \cup \{p\})$ using Equation 2 and Corollary 3.3;
return $\frac{PR(p, G(Farm \cup \{p\}))}{PR(p, G)}$;

Fig. 4. A local greedy search algorithm to extract (θ, k)-farms.

links pointing to p) as the candidates of page farm members. It greedily picks the page with the highest contribution among those in the candidate set, and adds the page into the page farm. Once a new page q is added into the farm, all those immediate neighbors of q (that is, those pages having links pointing to q) are added into the candidate set if their distances to p are no more than k. The selection continues iteratively until a farm contributing to a portion of at least θ of the PageRank of p is found, or the candidate set is empty. In the latter case, all the k-neighbors of p contribute to less than a θ portion of the PageRank of p.

In this algorithm, we do not require that the complete Web graph is available. Most search engines can return an approximate value of pages satisfying a specific keyword query. The number of pages in the Web graph, that is, N in the PageRank formula, can be estimated by submitting some simple queries (e.g., a search for the term a). Moreover, we can use services of search engines to extract (θ, k)-farms. For example, the Google "Search Links" operator returns some pages pointing to a target page p directly. Although the current inlink search services provided by those major search engines cannot return all inlinks, our method still can capture the spamicity of a page without a whole Web graph. With better inlink search results (as those major search engine companies claim to provide in the future), our detection results can be improved further.

In the local greedy search algorithm, only those pages whose distance to p is no more than k may be searched. Moreover, each of such pages can be included into the candidate set at most once. It never computes PageRank scores by an iterative converging method. Equation (2) and Corollary 3.3 help to speed up the computation of the PageRank contribution. First, the computation of contributions can be decomposed into computing the contributions from paths. Thus, when a new page is added to the page farm, we do not need to compute the contribution again completely. Instead, we can compute the incremental part using Corollary 3.3. Second, once a page is added into the page farm, the contributions of the pages in the candidate set can be updated accordingly. Therefore, it is much more efficient than the naïve greedy search algorithm.

We also consider another naïve method. In order to calculate the page contribution from page q to page p, instead of using path contributions to calculate more accurate page contribution, we use $\frac{PR(q,G)}{OutDeg(q)}$ as a rough estimation. Though the algorithm is simpler and more efficient, the results turn out to be unsatisfactory. There are several reasons. First, the heuristic that a page of larger value of $\frac{PR(q,G)}{OutDeg(q)}$ may have larger page contribution does not necessarily hold. Second, when we expand the farm, we consider the pages with a distance to the target page up to k. The fraction $\frac{PR(q,G)}{OutDeg(q)}$ can only reveal the direct connections between the two pages, but not the longer link paths between the two pages. As a conclusion, our local greedy search algorithm can balance efficiency and accuracy well.

Figure 4 shows the algorithm to extract the (θ, k)-farm for a specific target page. The algorithm can be extended to extract the (θ, k)-farms for a set of pages easily. A straightforward way is as follows. We run the algorithm for pages one by one. Once the farm for a target page is extracted, we can consider the next target page which is pointed to by the current target page. Thus, some path contributions calculated for the current target page can be reused or updated for the next page. For example, consider the graph shown in Figure 3. Suppose the farm of page p_1 has been extracted. We can consider p as the next target page, since p is directly pointed by p_1. When calculating the page contribution and path contribution, the previous results can be reused. Specifically, the path contributions from other pages through p_1 to p can be easily obtained from the path contribution from those pages to p_1 in the previous step, since the lengths of all the paths are increased by 1 only.

Our greedy method extracts page farms one by one. As will be shown in Figure 7, our method has a linear scalability empirically. However, extracting page farms for all pages on the Web is still time-consuming.

In many cases, a user may be interested in whether some specific Web pages are spam target pages. Our method can be applied to such individual pages on the fly. Only the page farms of those pages need to be extracted and analyzed. We only need to search the neighbors of those pages, which is highly feasible.

4. LINK SPAM DETECTION

Driven by the huge potential benefit of promoting rankings of Web pages, many attempts have been made to boost page rankings by making up some artificially designed link structures, which are known as *link spam* [Bianchini et al. 2005; Gyöngyi and Garcia-Molina 2005b; Henzinger et al. 2003; Langville and Meyer 2004].

Definition 4.1 (*Link Spam Target Page*). A link spam target page refers to a target page that benefits from any deliberate human action that is meant to trigger an unjustifiably favorable link-based relevance (that is, how the page is related to the search query) or importance (that is, how the page is ranked relative to the other pages on the Web) for some Web page comparing to the true value of the page.

So far, the techniques of Web spam can be classified into two categories, *term spam* and *link spam* [Gyöngyi and Garcia-Molina 2005b]. Term spam is to inject into a Web page many (irrelevant) keywords, which are often visually hidden, so that the page can be retrieved by many search queries that may be semantically irrelevant to the page. Link spam is to deliberately build auxiliary pages and links to boost the PageRank or other link-based ranking score of the target Web page. Due to the extensive adoptions of the link-based ranking metrics such as PageRank [Page et al. 1998] and HITS [Kleinberg 1999] in the popular Web search engines, link spam has been used deliberately by many spam pages on the Web.

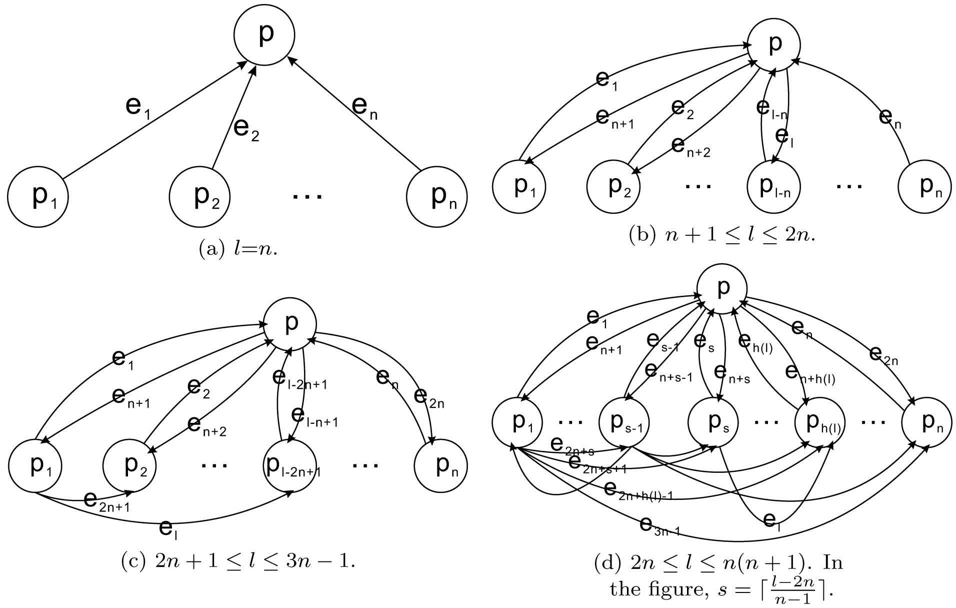
Some previous studies [Fetterly et al. 2004; Ntoulas et al. 2006; Castillo et al. 2007] treat spam detection as a traditional classification problem. Each page is assigned a label, either spam or not. However, the judgement on whether a page is spam or not, to some extent, is subjective. As improving the significance and the impact of a Web page is quite often a natural intent of the Web page builder, it is critical to measure whether the page is intended to be built with an unjustifiable high ranking score and the degree of such deliberations. The modeling of the degree of deliberation, which refers to "spamicity," becomes an essential role to spam detection.

In this section, we develop link spam detection methods using page farms. The experimental results to be shown in Section 5.1 will show that page farms have some potentials for link spam detection. First, the size of a page farm is relatively small comparing to the whole Web. Thus, extracting page farms for a large set of pages is potentially feasible in practice. As will be shown in our experiments, on average the page farm of a Web page contains a small number of Web pages, which can be extracted without too much effort using the algorithm in Figure 4. Second, different Web pages have their own page farms. Those page farms may be quite different with respect to the link structures, thus it may be possible to use page farms to detect link spam target pages, since their page farms may be quite different from others.

The general idea is that we can calculate a spamicity score from the page farm of a Web page to measure the likelihood of the page being a link spam target page. We explore two alternatives of defining spamicity.

In order to judge whether a page benefits from link spam, we only need to extract the page farm of the target page. As shown in our experiments, on average the page farm of a Web page contains a small number of Web pages, which can be extracted without too much effort using the algorithm in Figure 4.

The induced subgraphs of page farms are directed graphs consisting of Web pages and links. For the sake of brevity, we use $Farm(p) = (V, E)$ to refer to the induced subgraph of the page farm of p, and do not distinguish a page farm and its induced subgraph.



(a) $l{=}n$.

(b) $n+1 \leq l \leq 2n$.

(c) $2n+1 \leq l \leq 3n-1$.

(d) $2n \leq l \leq n(n+1)$. In the figure, $s = \lceil \frac{l-2n}{n-1} \rceil$.

Fig. 5. Achieving the maximum PageRank scores.

4.1 Utility-based Spamicity

If p is the target of link spam, then $Farm(p)$ should try to achieve the PageRank of p as high as possible. We can calculate the maximum PageRank using the same number of pages and the same number of hyperlinks as $Farm(p)$ has. The utility of the page farm of p is the ratio of the PageRank of p against the maximum PageRank that can be achieved. The utility can be used as a measure on the likelihood that p benefits from link spam. Intuitively, if the utility is closer to 1, the page is more likely to be a link spam target page.

Then, what is the largest PageRank score that a farm of n pages and l links can achieve?

THEOREM 4.2 (MAXIMUM PAGERANK SCORES). *Let p be the target page, and $Farm(p)$ contain pages $p_1, \ldots, p_n$ $(p \neq p_i, i = 1, \ldots, n)$ and hyperlinks $e_1, \ldots, e_l$, $l \geq n$. The following structure maximizes the PageRank score of p.*

$$e_j = \begin{cases} p_j \rightarrow p & (1 \leq j \leq n) \\ p \rightarrow p_{j-n} & (n+1 \leq j \leq 2n) \\ p_{\lceil \frac{j-2n}{n-1} \rceil} \rightarrow p_{h(j)} & (2n+1 \leq j \leq l) \end{cases}$$

where $h(j) = 1 + (j - 2n - \lceil \dfrac{j-2n}{n-1} \rceil (n-2) + 1) \bmod n$.

PROOF. In order to connect every page in the farm to the target page, at least n links are needed. Thus, $l \geq n$. The structure can be divided into four cases.

First, when $l = n$, then $e_j = p_j \rightarrow p$, as shown in Figure 5(a).

Second, when $n+1 \leq l \leq 2n$, then, as shown in Figure 5(b),

$$e_j = \begin{cases} p_j \to p & (1 \leq j \leq n) \\ p \to p_{j-n} & (n+1 \leq j \leq l) \end{cases}$$

Third, as shown in Figure 5(c), when $2n+1 \leq l \leq 3n-1$,

$$e_j = \begin{cases} p_j \to p & (1 \leq j \leq n) \\ p \to p_{j-n} & (n+1 \leq j \leq 2n) \\ p_1 \to p_{j-2n+1} & (2n+1 \leq j \leq l) \end{cases}$$

Last, When $3n \leq l \leq n(n+1)$, then the structure is as shown in Figure 5(d).

Recall the path contribution in Definition 3.2; given a link path P from v_1 to p: $v_1 \to v_2 \to \ldots \to v_n \to p$,

$$LCont(P, p) = \frac{1}{N} d^{|P|}(1-d) \prod_{i=1}^{n} \frac{1}{OutDeg(v_i)}.$$

Thus, $LCont(P, p)$ is based on two factors: the length of the link path $|P|$ and the contribution propagation $\pi(P) = \prod_{i=1}^{n} \frac{1}{OutDeg(v_i)}$. When $d = 1$, $LCont(P, p) = 0$, and it is a trivial case. In the following analysis, let us assume $d < 1$. Intuitively, the smaller the length of the link path, the larger the path contribution. Moreover, the larger the contribution propagation on the link path, the larger the path contribution.

Based on the above analysis, we next prove the optimum of each case one by one.

We first show the optimum of case (a) by an induction on l, that is, the number of links in the farm. As the basis step, when $l = 1$ (thus $n = 1$), page p_1 should have a link pointing to p. It is the optimal structure. As an inductive step, suppose when $l = j$, the optimal structure is shown in Figure 5(a). We want to construct the optimal structure when $l = j + 1$. Since only one new edge is available, it is obvious that each page p_j $(1 \leq j \leq n)$ should point to p directly. That is, for any page p_j, there is a link $p_j \to p$. In this case, the link path P from p_j to p has the smallest length $|P| = 1$ and the largest contribution propagation $\pi(P) = 1$, thus the largest path contribution. As a result, the optimal structure when $l = j + 1$ is shown in Figure 5(a). From the basis step and the inductive step, we can conclude that the structure in Figure 5(a) is the optimal structure when $l = n$.

We next show the optimum of case (b) by an induction on l as well. (1) As the basis step, we consider when $l = n+1$. When the number of edges is larger than the number of pages, each page p_j $(1 \leq j \leq n)$ can contribute most to p if there is a circle between p_j and p. That is, for any page p_j, there are two links $p_j \to p$ and $p \to p_j$. In this way, there are infinite link paths from p_j to p, thus p obtains the maximum contribution from p_j. Please note that creating a loop between p_j and p_{j-1} does not make p_j contribute most to p. According to the formula of path contribution, by creating a loop between p_j and p_{j-1}, the PageRank contribution from p_j largely goes into p_{j-1}. Our main goal is to increase the PageRank score of p as much as possible. Thus creating loops between pages p_j and p is with the highest priority. As a result, the optimal structure when $l = n + 1$ is shown in Figure 5(b). (2) As an inductive step, suppose when $l = j$ $(n < j < 2n)$, the optimal structure is shown in Figure 5(b). We want to construct the optimal structure when $l = j + 1$. This construction problem has the "greedy choice property" and "optimal substructure" [Cormen et al. 2001], that is, the optimal farm when $l = j + 1$ can be obtained by adding one more link to the optimal farm when $l = j$, such that the increase of the sum of the PageRank scores of p_j $(1 \leq j \leq n)$ is smallest. Otherwise, we can simply use the "cut and paste" method [Cormen et al. 2001] to obtain a better structure. Based on Corollary 3.3, by adding the new link from p to p_j, we can create the maximum number of link paths to p, thus p's PageRank score is increased most. As a result, the optimal structure when $l = j + 1$ is shown in Figure 5(b). From the basis step and the inductive step, we can conclude that the structure in Figure 5(a) is the optimal structure when $n+1 \leq l \leq 2n$.

We next show the optimum of case (c). Similarly, we follow an induction on l. (1) As the basis step, when $l = 2n+1$, there is one link from p_j to p_k. Obviously, any link from p_j to p_k has the same effect, thus we simply select $p_1 \rightarrow p_2$. So the structure in Figure 5(c) is the optimal structure when $l = 2n + 1$. (2) As an inductive step, suppose when $l = j$ $(2n + 1 \leq j \leq 3n - 2)$, the optimal structure is shown in Figure 5(c). We want to construct the optimal structure when $l = j + 1$. As proved in Page et al. [1998] and Langville and Meyer [2004], given a Web graph with n nodes, if each node has the out-degree at least 1, the sum of the PageRank scores of these n nodes in the Web graph is equal to n (unnormalized by the number of pages in the graph). So in case (c), the sum of the PageRank scores of p and p_j $(1 \leq j \leq n)$ is equal to $n + 1$. In order to maximize the PageRank score of p, we have to minimize the sum of the PageRank scores of p_j $(1 \leq j \leq n)$. Since the new link only can be added from p_j to p_k where $1 \leq j, k \leq n$, we want to increase the PageRank scores of p_k as little as possible, thus the decrease of the PageRank score of p is minimal. This objective can be achieved by adding the link from p_1 to p_{j-2n+1}, since the new link to p_{j-2n+1} has the length $|P| = 1$ and the smallest contribution propagation $\pi(P) = \frac{1}{j-2n+1}$. So the optimal structure when $l = j+1$ is as shown in Figure 5(c). From the basis step and the inductive step, we can conclude that the structure in Figure 5(c) is the optimal structure when $2n + 1 \leq l \leq 3n - 1$.

We observe that case (c) in Figure 5 is a special case for case (d). Thus, the optimum of Case (d) can be proved in the same way. Theorem 4.2 is proved. □

Please note that our Web graph model follows the traditional configuration. Self-loops are removed from the graph. Moreover, in our PageRank contribution model, we assume that the dangling nodes (which are generated by vertex voiding) can be kept unchanged in the induced graph, as mentioned in Section 2.

Based on Theorem 4.2, we denote by $PR_{max}(n, l)$ the maximum PageRank score that a page farm of n pages and l intra-links can achieve.

Moreover, we have the following corollary.

Corollary 4.3 (Maximum PageRank Scores ($n \leq l \leq 2n$)). *In Figure 5, the maximum PageRank score $PR_{max}(n, l)$ in cases (a) and (b) is given by*

$$PR_{max}(n,l) = \begin{cases} \frac{(dn+1)(1-d)}{N} & (l = n) \\ \frac{nd+1}{N(1+d)} & (n < l \leq 2n) \end{cases}$$

Proof. The proof can be constructed by solving the sytem of PageRank equations for all the pages. Here, we give a proof using path contributions to elaborate the use of path contribution in PageRank calculation.

We first show the case when $l = n$. The optimal structure is shown in Figure 5(a). The way to calculate the maximum PageRank score in Case (a) is as follows: there are totally n link paths to the target page p. Based on Equation (2), we have

$$PR(p) = \frac{1-d}{N} + \sum_{k=1}^{n} LCont(e_k, p).$$

For each Path contribution $LCont(e_k, p)$, according to Definition 3.2, we have

$$\begin{aligned} LCont(e_k, p) &= \frac{d(1-d)}{N}\frac{1}{1} \\ &= \frac{d(1-d)}{N}. \end{aligned}$$

So

$$\begin{aligned} PR_{max}(n,l) &= \frac{1-d}{N} + n\frac{d(1-d)}{N} \\ &= \frac{(dn+1)(1-d)}{N}. \end{aligned}$$

We next show the case when $n < l \leq 2n$. The optimal structure is shown in Figure 5(b). The way to calculate the maximum PageRank score in Case (b) is as follows.

In Figure 5(b), the value k is equal to $l - n$. So there are totally k pages having a link pointed by p, and $n - k$ pages having no links pointed by p. According to Equation (2), we have to find all the different link paths pointing to the target page p, calculate the path contributions, and then sum them up. We can classify all the link paths by their lengths into the following categories: paths with length i, where $i = 1, 2, \ldots$.

We define some notations first. A link path can be denoted as $p_1 \to p_2 \to \ldots \to p_n$, where $p_1, p_2, \ldots, p_n$ are the pages on this path. We use $\{p_1, p_2, \ldots, p_n\}$ to denote a set of pages. For simplicity, we use $p_1 \to \ldots \to \{p_j, \ldots, p_k\} \to \ldots \to p_n$ to denote any path $p_1 \ldots \to p_m \to \ldots \to p_n$ where $p_m \in \{p_j, \ldots, p_k\}$. We use $\mathcal{PATH}_i$ to denote the set of link paths pointing to p with length i, where $i = 1, 2, \ldots$. We use $LCont(\mathcal{PATH}_i)$ to denote the total path contributions where the paths are in the set of $\mathcal{PATH}_i$. For a path $P \in \mathcal{PATH}_i$, we use $LCont(P)$ to denote the path contribution of P. Now we prove

$$LCont(\mathcal{PATH}_{i+2}) = d^2 \times LCont(\mathcal{PATH}_i).$$

Considering the optimal structure shown in Figure 5(b), for any path $P \in \mathcal{PATH}_i$, we can easily obtain a path $P \in \mathcal{PATH}_{i+2}$, which can be constructed by adding two new links $p \to p_m \to p$ to the end of $P \in \mathcal{PATH}_i$, where $p_m \in \{p_1, p_2, \ldots, p_k\}$. Thus, given a path $P \in \mathcal{PATH}_i$, we can get k paths $P \in \mathcal{PATH}_{i+2}$. According to Definition 3.2, we have

$$\begin{aligned} LCont(\mathcal{PATH}_{i+2}) &= \sum_{P \in \mathcal{PATH}_{i+2}} LCont(P) \\ &= \sum_{P \in \mathcal{PATH}_i} k \times LCont(P) \times \frac{d^2}{k} \\ &= \sum_{P \in \mathcal{PATH}_i} d^2 \times LCont(P) \\ &= d^2 \times LCont(\mathcal{PATH}_i). \end{aligned}$$

Thus, we have

$$\begin{aligned} PR_{max}(n, l) &= \frac{1-d}{N} + \sum_{i=1}^{\infty} LCont(\mathcal{PATH}_i) \\ &= \frac{1-d}{N} + \sum_{i=0}^{\infty} LCont(\mathcal{PATH}_{2i+2}) + + \sum_{i=0}^{\infty} LCont(\mathcal{PATH}_{2i+1}) \\ &= \frac{1-d}{N} + \sum_{i=0}^{\infty} d^{2i} LCont(\mathcal{PATH}_2) + + \sum_{i=0}^{\infty} d^{2i} LCont(\mathcal{PATH}_1) \\ &= \frac{1-d}{N} + (LCont(\mathcal{PATH}_1) + LCont(\mathcal{PATH}_2)) \times \sum_{i=0}^{\infty} d^{2i} \\ &= \frac{1-d}{N} + \frac{1}{1-d^2} \times (LCont(\mathcal{PATH}_1) + LCont(\mathcal{PATH}_2)). \end{aligned} \tag{5}$$

$LCont(\mathcal{PATH}_1)$ and $LCont(\mathcal{PATH}_2)$ can be calculated as the following two cases. In the first case, for paths of length 1, we have two types of paths.

—For paths $p_i \to p$ $(1 \leq i \leq k)$, the sum of path contribution is

$$\sum LCont = \frac{d(1-d)}{N} \times k.$$

—For paths $p_i \to p$ $(k < i \leq n)$, the sum of path contribution is

$$\sum LCont = \frac{d(1-d)}{N} \times (n-k).$$

In the second case, for the path of length 2, that is $p \to p_i \to p$ $(1 \leq i \leq k)$, the sum of path contribution is

$$\sum LCont = d^2(1-d)\frac{1}{Nk} \times k.$$

Thus, we have

$$LCont(\mathcal{PATH}_1) = \frac{kd(1-d)}{N} + \frac{d(1-d)(n-k)}{N} = \frac{nd(1-d)}{N} \tag{6}$$

and

$$LCont(\mathcal{PATH}_2) = \frac{d^2(1-d)}{N}. \tag{7}$$

We apply Equation (6) and Equation (7) to Equation (5), then we have

$$PR_{max}(n,l) = \frac{1-d}{N} + \frac{1}{1-d^2} \times (\frac{nd(1-d)}{N} + \frac{d^2(1-d)}{N}) = \frac{nd+1}{N(1+d)}. \tag{8}$$

□

Corollary 4.3 gives the maximum PageRank scores for Cases (a) and Case (b) in Figure 5 directly. However, for the other cases, there are no simple and direct ways to calculate the exact maximum PageRank scores. In our implementation, we construct the optimal structure graphs first, and then compute the maximum PageRank scores.

A page farm of n pages and l hyperlinks is called an *optimal spam farm* if the target page achieves the maximum PageRank score.

Definition 4.4 (*Utility-Based Spamicity*). For a target page p, let $Farm(p) = (V, E)$ be the page farm of p. We define the *utility-based spamicity* of p as

$$USpam(p) = \frac{PR(p)}{PR_{max}(|V|, |E|)},$$

where $PR(p)$ and $PR_{max}(|V|, |E|)$ represent the PageRank score of p based on $Farm(p)$.

The utility-based spamicity of a Web page is between 0 and 1. The higher the utility-based spamicity, the more the page farm is utilized to boost the PageRank of the target page. The spammers (that is, the builders of spam Web pages) build up the "link spam farms" with the only purpose to boost the rankings of the target pages as much as possible. The optimal link spam farms do not commonly happen on the Web, because they are quite different from those normal page farms.

Moreover, since optimal link spam farms are highly regular as indicated by Theorem 4.2, a search engine may easily detect the optimal link spam farms. To disguise, a spammer may modify the optimal link spam farm but still keep the target pages of high PageRank scores. Using the utility-based spamicity to detect link spam, we can still capture the disguised link spam since the disguised link spam still has a page farm efficiently boosting the PageRank score of the target page.

There are several critical differences between the results in this section and those in Bianchini et al. [2005].

First, the general goals of our work and that of Bianchini et al. [2005] are different. Bianchini et al. [2005] focus on analyzing the influence of dangling nodes to achieve the optimal PageRank score of a target page. Our work focuses on finding how to construct the link structures among several pages in the farm to achieve the optimal PageRank scores.

Second, the general problem settings of our work and that in Bianchini et al. [2005] are different. Bianchini et al. [2005] consider only the number of pages as the optimization parameter. In our work, we consider not only the number of pages in the farm, but also the number of hyperlinks among them.

Third, the technical results of our work and that in Bianchini et al. [2005] are different. Bianchini et al. [2005] only show that the link structure of Case (b) in Theorem 5.2 is the optimal structure for the target page to achieve the highest PageRank score. Moreover, in Bianchini et al. [2005], the optimality of Case (b) was proved using a different method. Our work is more comprehensive in the sense that we consider different cases and show the optimal structure for each case. We give the complete proof for all the cases, including case (b), for the sake of completeness.

4.2 Characteristics-Based Spamicity

Since a page farm captures the most significant contributors to the PageRank score of the target page and the link structures, we can examine the characteristics of the page farm to evaluate the likelihood of link spam for the target page. In this section, we identify three heuristics to measure the likelihood of link spam for a Web page.

4.2.1 *Contributor PageRank Heuristic.* As indicated by the studies on authoritative pages and hub pages [Kleinberg 1999], a Web page is semantically important if it is pointed to by some authoritative pages or hub pages, which often have high PageRank scores. Intuitively, a link spam farm tend to have a "flatter" distribution than naturally emerging structures, which are closer to a power law distribution. Heuristically, if a page has a high PageRank score but its page farm does not have any page of high PageRank score, then it is likely the page is a link spam target page.

Based on this idea, we can measure the difference of the PageRank score of the target page and the average score of its page farm. Technically, we define the PageRank boosting ratio to measure the difference.

Definition 4.5 (*PageRank Boosting Ratio*). The *PageRank boosting ratio* is the ratio of the PageRank of p against the average PageRank of pages in $Farm(p) = (V, E)$. That is,

$$\beta(p) = \frac{PR(p)}{\frac{1}{|V|}\sum_{p' \in V} PR(p')}.$$

HEURISTIC 1 (CONTRIBUTOR PAGERANK). *The larger the PageRank boosting ratio, the more likely a page is a link spam target page.*

Benczur et al. [2005], suggested examining the PageRank distributions among some other pages linking to the target page. However, their idea is different from ours. First, they suggested to extract the supporter pages to a target page using Monte Carlo simulation, which may not be those important pages contributing most to the target page. Second, they suggested to penalize the PageRank score of the target page according to the PageRank distributions of those support pages. Our method does not explicitly examine the PageRank distribution of pages in the farm, but examines whether the PageRank score of the target page is well boosted.

4.2.2 *Link Efficiency Heuristic.* From Theorem 4.2, we have the following result, which is a more general result than the optimal structure given in Gyöngyi and Garcia-Molina [2005a]. In Gyöngyi and Garcia-Molina [2005a], the number of links is confined to range $[n+1, 2n]$, where n is the number of pages.

COROLLARY 4.6. *For a target page p whose page farm has n pages, $PR(p) \leq \frac{nd+1}{N(1+d)}$. The maximum PageRank score is achieved when there are l $(n+1 \leq l \leq 2n)$ hyperlinks in the farm as configured in Theorem 4.2.*

PROOF. As shown in Corollary 4.3, for $l = n$, $PR_{max}(n, l) = \frac{(dn+1)(1-d)}{N}$; for $n < l \leq 2n$, $PR_{max}(n, l) = \frac{nd+1}{N(1+d)}$. When l increases, more links need to be added into the graph. However, those links will distract some contributions to the other pages in the farm, thus the maximum PageRank scores in Cases (c) and (d) shown in Figure 5 are less than that in Case (b). As a result, given n pages in the farm, Case (b) is the optimal structure. □

A page farm of n Web pages must have at least n hyperlinks to connect each page in the farm to the target page. Based on Corollary 4.6, the more hyperlinks in the page farm, the less efficiently those links are used to boost the PageRank of the target page. We define the link efficiency of a page farm to capture this feature.

Definition 4.7 (*Link Efficiency*). The *link efficiency* of the farm is the ratio of the number of pages in $Farm(p) = (V, E)$ against the total number of links between the pages in V. That is,

$$\iota(p) = \frac{|V|}{|\{p_1 \to p_2 \in E | p_1 \neq p, p_2 \neq p\}|}.$$

It is worth noting that there may not exist edges among pages in the farm (e.g., Cases (a) and (b) in Figure 5). In this case, $\iota(p)$ is $+\infty$. The farms are optimal link spam farms. In general, $\iota(p)$ is not bounded in range [0, 1].

In an average page farm that is not for link spam, a few arbitrary hyperlinks may exist between pages in the farm. On the other hand, in order to fully boost the target page, pages in a link spam farm often do not point to each other. Based on this observation, we have the following link efficiency heuristic.

HEURISTIC 2 (LINK EFFICIENCY). *The larger the link efficiency, the more likely a page is a link spam target page.*

4.2.3 *Centralization Heuristic.* In an ideal link spam farm, the target page has a large indegree, since hyperlinks point to the target page from the pages in the farm. The pages in such a farm often have low indegree since otherwise the efficiency of the pages and the links in the page farm is reduced. In other words, the links and the pages in a link spam farm are highly centralized such that the target page is at the center of the farm. We measure the centralization degree using this hint.

Definition 4.8 (*Centralization Degree*). The *centralization degree* of the farm is the ratio of the indegree of p against the average indegree of the pages in $Farm(p) = (V, E)$. That is,

$$\kappa(p) = \frac{InDeg(p)}{\frac{1}{|V|}\sum_{p' \in V} InDeg(p')},$$

where $InDeg(p)$ and $InDeg(p')$ represent the indegrees of page p and p' in $Farm(p)$.

HEURISTIC 3 (CENTRALIZATION DEGREE). *The larger the centralization degree, the more likely a page is a link spam target page.*

4.2.4 *Characteristics-Based Spamicity.* Consider a virtually non-spam page p and its page farm $Farm(p) = (V, E)$. We have the following heuristics.

—Ideally, a virtually nonlink spam target page is not boosted by a large collection of pages of very low page rank. Thus, the PageRank boosting ratio $\beta(p)$ should be a small number, virtually approaching 1 when the PageRank of p is not boosted.

—Since page p is not boosted by any authoritative or hub pages, in order to achieve some nontrivial PageRank score, the page farm $Farm(p)$ has to contain many pages. At the same time, a random hyperlink $p_i \rightarrow p_j$ has the constant probability $1 - d$ (the random jump probability) for $p_i, p_j \in V$. Therefore,

$$\iota(p) = \lim_{n\to\infty} \frac{n}{p^{\frac{n(n-1)}{2}}} = 0.$$

—The centralization degree $\kappa(p)$ of the page farm should approach 1, since the probability that a page $p' \neq p$ links to p directly is the same as the probability that p' links to any other pages in the farm.

The three different heuristics proposed above capture three different aspects: PageRank scores, hyperlinks and degrees in the farm. Based on the previous observations, we define the characteristics-based spamicity as follows.

Definition 4.9 (*Characteristics-Based Spamicity*). For page p, the *characteristics-based spamicity* is

$$CSpam(p) = \sqrt[\gamma]{|\beta(p) - 1|^\gamma + |\iota(p)|^\gamma + |\kappa(p) - 1|^\gamma},$$

where $\gamma > 0$ is the *Minkowski distance parameter* [Thompson 1996].

Please note that each of these three heuristics may not work for all Web pages. However, combining the three heuristics may work well for many Web pages, as verified by our experimental results in Section 5.

5. EXPERIMENTAL RESULTS

In this section, we report a series of experimental results on link spam detection using page farms. All the experiments were conducted on a PC computer with a 3.0 GHz Pentium 4 CPU, 1.0 GB main memory, and a 160 GB hard disk, running the Microsoft Windows XP SP2 Professional Edition operating system.

The programs were implemented in C/C++ using Microsoft Visual Studio .NET 2003.

We used the recently released webspam-uk2006 dataset by Yahoo! Research Barcelona[2]. The dataset [Castillo et al. 2006] is the result of the effort of an international team of volunteers. As far as we know, the webspam-uk2006 dataset is the first publicly available benchmark for Web spam detection.

The *base dataset* contains 77, 862, 535 pages in the domain of .UK downloaded in May 2006 by the Laboratory of Web Algorithmics, Universitá degli Studi di Milano. The *spam test collection* dataset consists of 8, 415 different hosts chosen from the *base dataset*. A team of volunteers was asked to classify this set of hosts as "normal," "spam," or "borderline." Moreover, the project organizers added two kinds of special votes: all the UK hosts that were mentioned in the Open Directory Project[3] in May 2006 are voted "normal," and all the UK hosts ending in .ac.uk, .sch.uk, .gov.uk, .mod.uk, .nhs.uk or .police.uk are voted "normal," too.

Whether a page is spam is labeled by assigning 1 point to each vote of "spam," 0.5 point to each vote of "borderline," and 0 point to each vote of "normal." The final label for a host is determined by the average of points from all votes on this host: an average of over 0.5 point is "spam," an average of less than 0.5 point is "normal," and an average of 0.5 point is "undecided." As a result, among 8, 415 different hosts, 7, 472 hosts are labeled as "normal," 767 hosts are labeled as "spam," and the remaining 176 hosts are labeled as "undecided." Since those "undecided" pages are borderline pages, it is hard to dogmatically give a label either "spam" or "normal." For simplicity, we removed those borderline hosts in the experiments.

Some Web pages in the dataset are labeled by the human experts and identified by our methods as spam pages, but are still indexed by some major search engines, such as `http://we-sell-it.co.uk/`, `http://www.uk-shop-uk.co.uk/`, `http://www.courses-on-line.co.uk`, and `http://www.findone.co.uk`.

We constructed the Web graph using the pages in the *base dataset* and computed the PageRank scores of the pages at the page level.

5.1 Extracting Page Farms

For the pages in the *spam test collection*, we extracted the (θ, k)-farms. To understand the effects of the two parameters θ and k on the page farms extracted, we used different values of θ and k, and measured the average size of the extracted farms. Figure 6 shows the results.

When k is very small (1 or 2), even selecting all pages of distance up to k may not be able to achieve the contribution threshold θ. Therefore, when k increases, the average page farm size increases. However, when k is 3 or larger, the page farm size does not increase much when θ increases. This verifies our observation that the near neighbor pages contribute more than the remote ones, and the PageRank of a page is mainly determined by its near neighbors.

[2]`http://aeserver.dis.uniroma1.it/webspam`
[3]`http://www.dmoz.org`

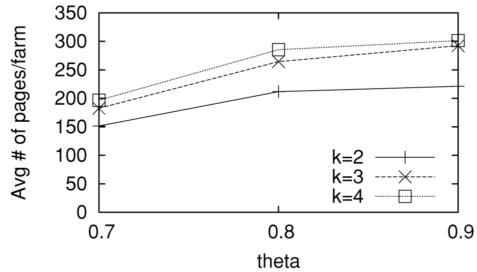


Fig. 6. The effects of parameters θ and k on page farm extraction.

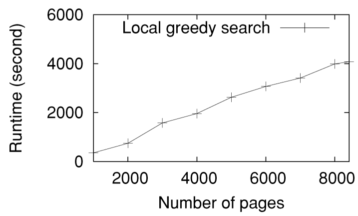


Fig. 7. The scalability of page farm extraction.

When θ increases, more pages are needed to make up the contribution ratio. However, the increase of the average page farm size is sublinear. The reason is that when a new page is added to the farm, the contributions of some pages already in the farm may increase due to the new paths from those pages to the target page through the new page. Therefore, a new page often boosts the contributions from multiple pages in the farm. Intuitively, the larger and the denser the farm, the more pages whose contributions can increase by adding a new page. On average, when $\theta \geq 0.8$, page farms are quite stable and capture the major contribution to PageRank scores of target pages.

We compared the page farms extracted using different settings of the two parameters. The farms are quite robust. That is, for the same target page, the page farms extracted using different parameters largely overlap. In the rest of this section, by default we report results on $(0.8, 3)$-farms of Web pages.

We tested the page farm extraction efficiency using the *naïve greedy search* algorithm and the *local greedy search* algorithm discussed in Section 3. We first computed the PageRank scores for the pages in the *spam test collection* using a simple power method, and then we used the Web graph to extract page farms for pages in the *spam test collection*. Each page is associated with a page farm. We show in Figure 7 the runtime of the local greedy search method with respect to the number of farms extracted. The runtime only includes the time for extracting page farms, and does not include the time for computing the PageRank scores in the whole graph. The local greedy search method scales linearly. The naïve greedy search method is much slower. Extracting one page using the naïve greedy search method on average needs $1,742$ seconds. Thus, the curve for the naïve greedy method is not plotted in the figure. As analyzed before, path contributions are much cheaper to compute. The average cost of extracting page farms using path contributions is much lower.

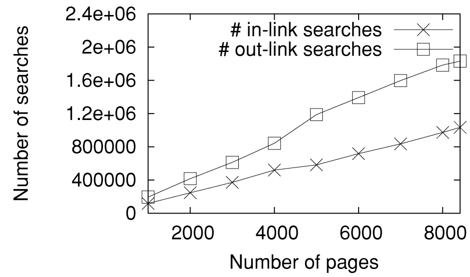


Fig. 8. Farm extraction cost in local greedy search.

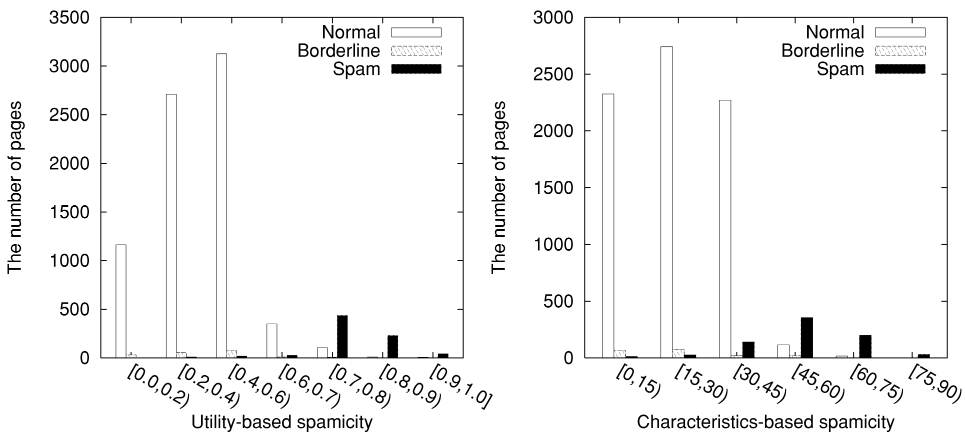


Fig. 9. The effectiveness of spamicity in spam detection.

To understand the cost of the local greedy search method in such a situation, in Figure 8, we tested the cost in the local greedy search method in two aspects. The number of in-link searches is the number of pages whose in-links are searched (for example, using the "Search Links" operator in Google). The number of out-link searches is the number of pages whose out-links are searched (for example, by parsing the page only for out-links). We counted the numbers of the in-link searches and out-link searches for each farm. The average numbers are comparable to the average number of pages in the page farms. This strongly suggests that the local greedy search method is efficient and scalable.

5.2 Spam Detection

To examine the effectiveness of the utility-based and characteristics-based spamicity measures, we compute the spamicity scores for the pages in the *spam test collection*. We group pages by their spamicity scores. Figure 9 shows the distribution of normal, borderline, and spam pages in groups with various ranges of spamicity scores. In the characteristics-based spamicity computation, we set the Minkowski distance parameter $\gamma = 2$ by default. When the spamicity is low, most pages are normal pages. When the spamicity is high, most pages are spam pages. Particularly, in this dataset, when the utility-based spamicity is over 0.7 and the characteristics-based spamicity is over 45, most pages

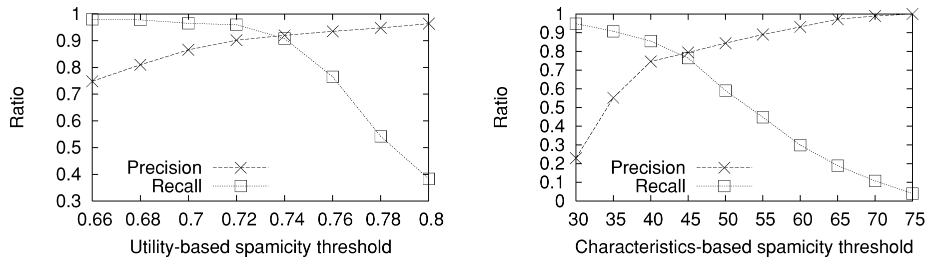


Fig. 10. The precision and the recall of utility-based and characteristics-based spamicity measures.

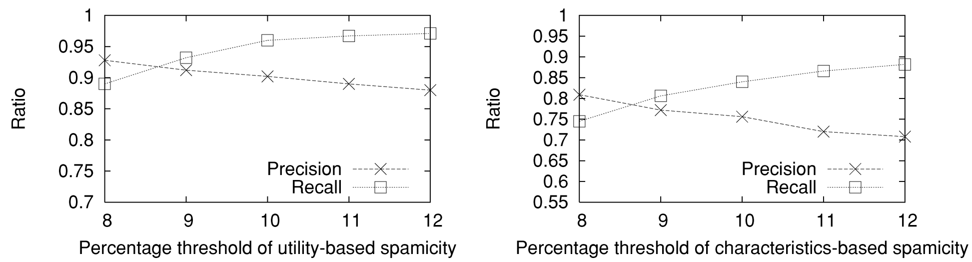


Fig. 11. The effectiveness of spam detection using a percentage threshold.

are spam. This set of experiments shows that the two spamicity measures can discriminate spam pages from normal ones.

We can simply set a spamicity threshold. The pages over the threshold are classified as spam, while the pages lower than the threshold are classified as normal. Figure 10 shows the precision and the recall of the two spamicity measures with respect to various spamicity threshold values. Generally, when the spamicity threshold goes up, fewer pages are detected as spam. The precision increases and the recall decreases. When the threshold is in the range of 0.7 to 0.74, the utility-based spamicity achieves the precision of more than 90% in detecting spam pages, and can catch more than 85% of the spam pages. When the threshold is in the range of 40 to 50, the characteristics-based spamicity has the precision and recall of more than 75%. The utility-based spamicity is more effective than the characteristics-based spamicity.

Alternatively, we can set a percentage threshold s and classify the top-s% pages having the highest spamicity scores as the suspect of spam pages, and the other pages as normal. Figure 11 shows the precision and the recall with respect to various percentage threshold values. Generally, as s increases, more pages are selected as spam pages. The precision decreases but the recall increases. When s is in the range of 8% to 9%, the precision and the recall of spam detection using utility-based spamicity is more than 90%. The detection using characteristics-based spamicity also achieves the best result in this range. This matches the ground truth (9.1% of the pages in this dataset are spam) well. In Figure 12, we compare the two spamicity methods using the F-measure. The utility-based spamicity is clearly more effective than the characteristics-based spamicity in detection quality.

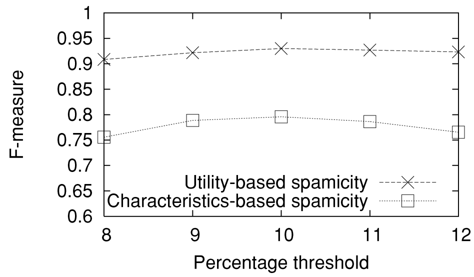


Fig. 12. The F-measure of the two spamicity methods.

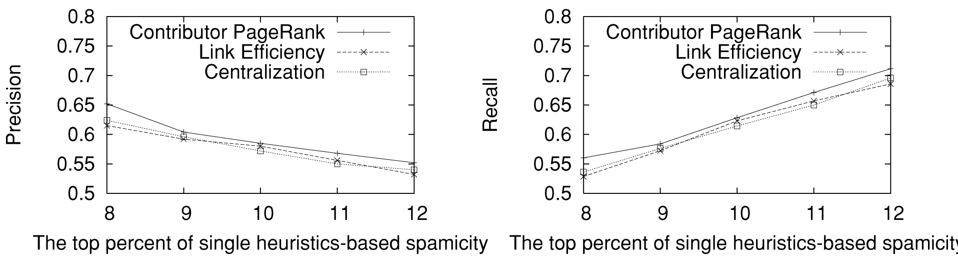


Fig. 13. The effectiveness of each heuristic in characteristics-based spamicity using a percentage threshold.

The F-measure results in Figure 12 are comparable to the results of the spam labeling challenge from the 2007 AIRWeb [Benczúr et al. 2007; Chien et al. 2007; Cormack 2007; Abou-Assaleh and Das 2007]. The winner method [Benczúr et al. 2007] is reported to achieve an F-measure score of 0.91 on average. This indicates that our methods are comparable in performance with the state-of-the-art spam detection methods such as Benczúr et al. [2007] and Chien et al. [2007]. The methods from AIRWeb 2007 do not use the spamicity-like approach. They need much more background information than our spamicity-based methods. For example, the winner method [Benczúr et al. 2007] used features including Microsoft OCI and Yahoo! Mindset, Google AdWords, and AdSense, as well as the graph similarity. Our method only needs the graph structure.

Since the characteristics based spamicity is based on three different heuristics, it is necessary to evaluate the effectiveness of each heuristic individually. Accordingly, we set a percentage threshold s and classify the top-$s\%$ pages having the highest scores as the suspect of spam pages. The spam detection precision and recall for each heuristic individually are shown in Figure 13. Clearly, individual heuristics do not work well in detecting spam pages. Comparing the results in Figure 11, the combination of three heuristics work much better.

We compared the utility-based spamicity method with SpamRank [Benczur et al. 2005], which is an existing method that detects link spam target pages by assigning a spamicity-like score and does not need supervised training. It assumes that spam pages have a biased distribution of pages that contribute to the undeserved high PageRank value. SpamRank penalizes pages that originate a suspicious PageRank share and personalizes PageRank on the penalties.

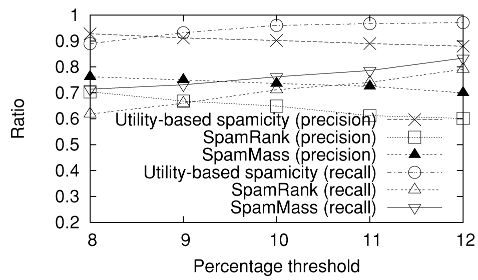


Fig. 14. The utility-based spamicity method and SpamRank.

We also compared our utility-based spamicity method with SpamMass [Gyöngyi et al. 2006]. SpamMass also uses a spamicity-like score to measure the degree of a page being a link spam target page. It is a well-known link spam detection method on the Web. The SpamMass approach is built on the idea of estimating the spam mass of nodes, which is a measure of the relative PageRank contribution of connected spam pages. Spam mass estimates are easy to compute using two sets of PageRank scores—a regular one and the other one with the random jump biased to some known good nodes.

A carefully chosen set of good nodes (the good core) is important for the success of SpamMass in link spam detection. We followed the core selection process described in Gyöngyi et al. [2006]. Three sets of hosts are selected to constitute the good core, in particular, the URLs that appear in Open Directory Project[4], the Web databases of educational institutions worldwide (`http://univ.cc/`), and those hosts ending with .ac.uk, .sch.uk, .gov.uk, .mod.uk, .nhs.uk or .police.uk.

We implemented the SpamRank and SpamMass methods as described in Benczur et al. [2005] and Gyöngyi et al. [2006], respectively. The results on the effectiveness are shown in Figure 14. The utility-based spamicity method clearly outperforms both SpamRank and SpamMass in terms of both precision and recall.

We also examined the efficiency of the different link spam target detection methods. In our utility-based spamicity method, the runtime includes the time for computing the PageRank scores in the whole graph, extracting page farms of target pages, and calculating the utility-based spamicity scores for target pages. In the SpamMass method, the runtime includes the time for two rounds of PageRank calculations in the whole graph, and calculating the SpamMass scores for target pages. In the SpamRank method, the runtime includes the time for one round of the Monte Carlo personalized PageRank calculations in the whole graph, and one round of PageRank calculations in the whole graph. The comparison is shown in Figure 15. To address the issue of extremely large Web graphs, we adopted the technique of *file mapping* which assigns a virtual address space to the Web graph file on the hard disk. In our experiments, the PageRank computation in the whole graph becomes a bottleneck. Among the

[4]Please note that some pages in Open Directory Project may be spam target pages. However, so far there are no efforts on cleaning the data completely, thus the good core may not be perfect.

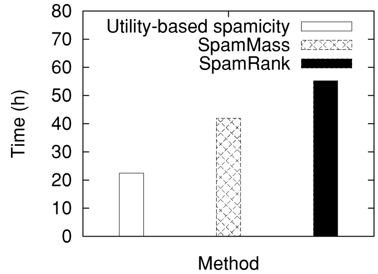


Fig. 15. The runtime of different link spam target detection methods.

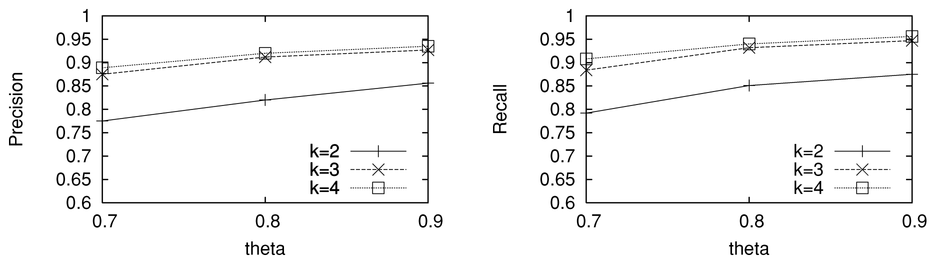


Fig. 16. The effect of θ on spam detection accuracy.

three methods, since our utility-based spamicity method needs only one round of PageRank calculation, it outperforms the other two methods.

We further examined the effects of page farms on the accuracy of spam detection. Since the utility-based spamicity is better than the characteristics-based spamicity, we only show the results on utility-based spamicity in Figure 16. We vary parameters θ and k, and classify the top 9% of the pages of the highest utility-based spamicity as the spam pages. We measure the precision and the recall of spam detection using different (θ, k)-farms. As discussed before, when θ and k increase, the page farms are more accurate. Thus the spam detection quality improves. Using $(\theta, 3)$- and $(\theta, 4)$-farms is much better than using $(\theta, 2)$-farms. The advantage of using $(\theta, 4)$-farms against using $(\theta, 3)$-farms is very small. Also, the quality is not very sensitive to θ when $\theta \geq 0.7$. This shows that whether a page is spam can be confidently determined using some near neighbors of the page.

5.3 Summary

The experiments clearly show that spam detection using page farms is feasible and effective. Our method outperforms SpamRank and SpamMass in both precision and recall. The utility-based spamicity is effective. Spam detection using utility-based spamicity can achieve high precision and high recall at the same time. Interestingly, when the *spam test collection* is formed, though some features come with the labels of neighborhood pages, the human volunteers made judgements mostly based on the content of the pages. In other words, those spam pages labeled in the dataset are typically adopted some term-based spam tricks as well. However, using the link analysis we can detect more than

90% of those spam pages. Since our method is focusing on detecting link spam tricks on the Web, this strongly indicates that most spam pages on the real Web use both link spam and term spam tricks.

Please note that, in Section 5.2, we try to find a threshold fitting the domain experts' judgement on spamicity and show the effectiveness of spam detection using our spamicity based methods. However, in practice, the spamicity threshold may be different for various users. A user can tune the threshold flexibly to reflect her/his tolerance of link spam. In the first place, the spamicity score of a Web page reflects the "degree" of the page being link spam. Such a score can be obtained by our methods without any training.

6. RELATED WORK

Our study is highly related to previous work in link-based ranking and Web spam detection. It is also related to social network analysis that has been studied extensively and deeply (see Wasserman and Faust [1994] and Scott [2000] as two textbooks). In this section, we only focus on some representative studies on link-based ranking and Web spam detection.

A few link-based ranking methods such as HITS [Kleinberg 1999] and PageRank [Page et al. 1998] were proposed to assign scores to Web pages to reflect their importance.

So far, Web page spam tricks can be divided into two categories, *term spam* and *link spam*. Gyöngyi et al. [2005b] referred link spam to the cases where spammers set up link structures of interconnected pages, called link spam farms, in order to boost the link-based ranking.

A single-target link spam farm model consists of three parts: a single target page to be boosted by the spammer, a (reasonable) number of boosting pages that deliberately improve the link-based ranking of the target page, and some external links accumulated from pages outside the link spam farm. Based on this model, given a fixed number of boosting pages, the optimal link structure which the target page can achieve the highest PageRank score is addressed in [Gyöngyi and Garcia-Molina 2005a]. However, their model does not consider the number of links between pages and thus it was unable to capture those disguised link spam. Gyöngyi et al. [2005a] also showed the link spam alliance which refers to the collaboration among spammers.

Some methods have been proposed to detect link spam. Fetterly et al. [2004] adopted statistical analysis to detect link spam. Several distribution graphs, such as the distribution of indegrees and outdegrees, were modeled well by some form of power law. A majority of the outliers were found to be spam by manually checking. Wu and Davison [2005] proposed an algorithm for link spam detection. It first generates a seed set of possible link spam farm pages based on the common link set between incoming and outgoing links of Web pages. Then, link spam pages are identified by expanding the seed set. Gyöngyi et al. [2006] introduced the concept of spam mass, a measure of the impact of link spam on a page's ranking. They discussed how to estimate spam mass and how the estimations can help to identify pages that benefit significantly from link spam. In Becchetti et al. [2006], studied the topology of the Web graph and computed

Web page attributes applying rank propagation and probabilistic counting over the Web graph. These attributes were then used to build a classifier.

Some other link spam detection methods resemble PageRank computation. For example, Benczur et al. [2005] proposed a method called SpamRank, which is based on the concept of personalized PageRank that detects pages with an undeserved high PageRank score. Gyöngyi et al. [2004] described an algorithm, called TrustRank, to combat Web spam. The basic assumption of TrustRank is that good pages usually point to good pages and seldom have links to spam pages. They first selected a bunch of known good seed pages and assigned high trust scores to them. Then, similar to PageRank, the trust scores were propagated via out-links to other Web pages. Finally, after convergence, the pages with high trust scores are believed to be good pages. However, TrustRank was vulnerable if the seed set used by TrustRank may not be sufficiently representative to cover well the different topics on the Web. In addition, for a given seed set, TrustRank has a bias towards larger communities. Wu et al. [2006] proposed the use of topical information to partition the seed set and calculate the trust scores for each topic separately to address the above issues. A combination of these trust scores for a page is used to determine its ranking.

In addition to link spam, term spam is another trick which is the practice of "engineering" the content of Web pages so that they appear relevant to popular searches. Most of the term spam detection methods use statistical analysis. For example, Fetterly et al. [2004], studied the prevalence of spam based on certain content-based properties of Web sites. They found that some features such as long host names, host names containing many dashes, dots and digits, as well as little variation in the number of words in each page within a site were good indicators of spam Web pages. Later, Fetterly et al. [2005] investigated the special case of "cut-and-paste" content spam, where Web pages were mosaics of textual chunks copied from legitimate pages on the Web. They also presented methods for detecting such pages by identifying popular shingles. Recently, Ntoulas et al. [2006] presented a number of heuristic methods for detecting content-based spam that essentially extend the previous work [Fetterly et al. 2004, 2005]. Some of those methods are more effective than the others; however, when used in isolation the methods may not identify all of the spam pages. Thus, Ntoulas et al. [2006] combines the spam-detection methods to create a highly accurate C4.5 classifier to detect term spam.

As described already, most of the previous studies focus on some graph structural properties which are highly associated with spam. At a very high level, our page farm based methods follow a similar philosophy. The utility based and characteristics based spamicity scores summarize the statistics of spam farms. However, our methods are essentially different from the previous work in the following aspects.

First, the link-based ranking methods and their applications do not analyze the environment of Web pages. In some link spam detection methods [Becchetti et al. 2006; Gyöngyi et al. 2006; Gyöngyi et al. 2004; Wu et al. 2006; Zhang et al. 2004], the concept of link spam farm is used to conceptually capture the set of Web pages that achieve the link spam. However, as far as we know, neither previous method nor empirical study have been proposed to extract link spam farms from the Web. Although in our page farm model, the page farm of a link spam target page is generally a superset of its link spam farm, the two farms are close to each other.

Second, our page farm model is different from the link spam farm model proposed in the previous studies. The previous work on link spam farms only addressed the concept and only considered some known link spam target pages. However, analyzing link spam farms in general is difficult because we may not know whether a target page is a link spam target page. Moreover, link spam farm extraction is a challenging problem. In our page farm model, each page has its own page farm. We can extract page farms for any page on the Web. We can distinguish link spam target pages from normal pages by examining their page farms. The page farm of a link spam target page can be used to obtain better understanding of its exact link spam farm.

Third, our proposed page farm consists of those most important contributor pages to the target page. We take into account the page contribution as the weight of the relativeness. Becchetti et al. [2006] proposed the concept of "supporter." They called page q a supporter of page p at distance k if the shortest path from q to p formed by links has length k. However, in this definition, each page has the same contribution to the target page which is not precise. Moreover, it may introduce some noisy information.

Fourth, we use the page farms, particularly the utility and the characteristics of the page farms, to detect link spam target pages. By doing so, we not only can detect the link spam, but also can capture how the link spam is attempted using the link spam farms. Some previous work [Becchetti et al. 2006; Du et al. 2007; Gyöngyi and Garcia-Molina 2005a] discussed the optimal structure for those link spam farms. However, they constrain the number of links with respect to the number of pages. In our model, we consider the general case. Our optimal link structure captures more information and it also can detect disguised link spam.

Last, social network analysis is often concerned with the global properties of a social network and the communities. To the best of our knowledge, there is no previous work from social network aspect analyzing the distribution of local structures, which can be captured using our page farm model proposed in the article.

7. CONCLUSIONS

Ranking pages is an essential task in Web search. Interesting problems, for a Web page p, include: which other pages are the major contributors to the ranking score of p are, and how the contribution is made. In this article, we studied the page farm mining problem and its application in link spam detection. We summarize our major contributions as follows.

—First, we studied the page farm mining problem. A page farm is a (minimal) set of pages contributing to (a major portion of) the PageRank score of a target page. We proposed the notions of θ-farm and (θ, k)-farm, where θ in $[0, 1]$ is a contribution threshold and k is a distance threshold. We studied the computational complexity of finding page farms, and show that it is NP-hard. Then we developed a greedy method feasible in practice to extract approximate page farms.

—Second, we investigated the application of page farms in link spam detection. We proposed two methods. First, we measured the utility of a page farm, that is, the "perfectness" of a page farm in obtaining the maximum PageRank score, and used the utility as an indicator of the likeliness of being a link spam target page. Second, we used the statistics of page farms as the indicator of the likeliness of being a link spam target page. Using the measures we can detect link spam target pages.

—Last, we evaluated our link spam detection methods using a newly available real dataset. The pages were labeled by human experts. The experimental results showed that our methods are effective in detecting spam pages.

The page farm-based link spam detection methods proposed in the article is comparable to some other methods such as SpamRank and SpamMass with respect to accuracy, however, the philosophy of our method is quite different from those. In particular, the purpose of our method is computing spamicity measures for single pages, possibly at browsing time. On the other hand, the PageRank derived methods such as SpamRank and SpamMass derive spam-scores for all pages in a large graph at the same time. As future work, we plan to develop more efficient algorithms for page farm extraction and analysis.

ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers and Dr. Gregory Piatetsky-Shapiro, the associate editor, for their insightful, constructive, and detailed comments on the previous versions of this article. We are also grateful to Kathleen Tsoukalas for her informative comments which helped to improve the quality of the article.

REFERENCES

ABOU-ASSALEH, T. AND DAS, T. 2007. Extension and propagation of manual and automatic Web spam scores. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*. ACM.

ADAMIC, L. A. 1999. The small world Web. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99)*, S. Abiteboul and A.-M. Vercoustre Eds. Springer-Verlag, 443–452.

ALBERT, R., JEONG, H., AND BARABASI, A.-L. 1999. The diameter of the world wide Web. *Nature 401*, 130–131.

BAEZA-YATES, R., BOLDI, P., AND CASTILLO, C. 2006. Generalizing pagerank: Damping functions for link-based ranking algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. ACM Press, 308–315.

BECCHETTI, L., CASTILLO, C., DONATO, D., LEONARDI, S., AND BAEZA-YATES, R. 2006. Using rank propagation and probabilistic counting for link-based spam detection. In *Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD06)*. ACM Press.

BENCZÚR, A., BÍRÓ, I., CSALOGÁNY, K., AND SARLÓS, T. 2007. Web spam detection via commercial intent analysis. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*. ACM, 89–92.

BENCZUR, A. A., CSALOGANY, K., SARLOS, T., AND UHER, M. 2005. Spamrank: Fully automatic link spam detection. In *Proceedings of the 1st International Workshop on Adversarial InformationRetrieval on the Web (AIRWeb'05)*.

BIANCHINI, M., GORI, M., AND SCARSELLI, F. 2005. Inside pagerank. *ACM Trans. Intern. Techn. 5,* 1, 92–128.

BRINKMEIER, M. 2006. Pagerank revisited. *ACM Trans. Intern. Techn. 6*, 3, 282–301.

BRODER, A., KUMAR, R., MAGHOUL, F., RAGHAVAN, P., RAJAGOPALAN, S., STATA, R., TOMKINS, A., AND WIENER, J. 2000. Graph structure in the Web. In *Proceedings of the 9th International Conference on World Wide Web (WWW'00)*. North-Holland Publishing Co., 309–320.

CASTILLO, C., DONATO, D., BECCHETTI, L., BOLDI, P., SANTINI, M., AND VIGNA, S. 2006. A reference collection for Web spam. *SIGIR Forum 40*, 2, 11–24.

CASTILLO, C., DONATO, D., GIONIS, A., MURDOCK, V., AND SILVESTRI, F. 2007. Know your neighbors: Web spam detection using the Web topology. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*. ACM, 423–430.

CHIEN, S., FETTERLY, D., MANASSE, M., NAJORK, M., AND NTOULAS, A. 2007. Microsoft silicon valley Web spam challenge entry. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*. ACM.

CORMACK, G. V. 2007. Content-based Web spam detection. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*. ACM.

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. 2001. *Introduction to Algorithms*. McGraw-Hill Higher Education.

DU, Y., SHI, Y., AND ZHAO, X. 2007. Using spam farm to boost pagerank. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*.

FETTERLY, D., MANASSE, M., AND NAJORK, M. 2004. Spam, damn spam, and statistics: Using statistical analysis to locate spam Web pages. In *Proceedings of the 7th International Workshop on the Web and Databases (WebDB'04)*. ACM Press, 1–6.

FETTERLY, D., MANASSE, M., AND NAJORK, M. 2005. Detecting phrase-level duplication on the world wide Web. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. ACM Press, 170–177.

GYÖNGYI, Z., BERKHIN, P., GARCIA-MOLINA, H., AND PEDERSEN, J. 2006. Link-spam detection-based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Databases (VLDB'06)*. ACM, 439–450.

GYÖNGYI, Z. AND GARCIA-MOLINA, H. 2005a. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Databases (VLDB'05)*. ACM, 517–528.

GYÖNGYI, Z. AND GARCIA-MOLINA, H. 2005b. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05)*.

GYÖNGYI, Z., GARCIA-MOLINA, H., AND PEDERSEN, J. 2004. Combating Web spam with trustrank. In *Proceedings of the 30th International Conference on Very Large Databases (VLDB'04)*. Morgan Kaufmann, 576–587.

HENZINGER, M., MOTWANI, R., AND SILVERSTEIN, C. 2003. Challenges in Web search engines. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*. 1573–1579.

KARP, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*.

KLEINBERG, J. M. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46*, 5, 604–632.

LANGVILLE, A. AND MEYER, C. 2004. Deeper inside pagerank. *Intern. Math. 1*, 3, 335–380.

NTOULAS, A., NAJORK, M., MANASSE, M., AND FETTERLY, D. 2006. Detecting spam Web pages through content analysis. In *Proceedings of the 15th International World Wide Web Conference (WWW'06)*. ACM Press, 83–92.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The pagerank citation ranking: Bringing order to the Web. Tech. rep., Stanford University.

SCOTT, J. 2000. *Social Network Analysis Handbook*. Sage Publications Inc.

THOMPSON, A. C. 1996. *Minkowski Geometry*. Cambridge University Press, Cambridge, UK.

WASSERMAN, S. AND FAUST, K. 1994. *Social Network Analysis*. Cambridge University Press, Cambridge, UK.

WU, B. AND DAVISON, B. D. 2005. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference (WWW'05)*. ACM Press, 820–829.

WU, B., GOEL, V., AND DAVISON, B. D. 2006. Topical trustrank: Using topicality to combat Web spam. In *Proceedings of the 15th International World Wide Web Conference (WWW'06)*. ACM Press, 63–72.

ZHANG, H., GOEL, A., GOVINDAN, R., MASON, K., AND ROY, B. V. 2004. Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web Graph (WAW'04)*. Lecture Notes in Computer Science, vol. 3243. Springer, 92–104.

Received May 2007; revised May 2008; accepted December 2008