MINING PAGE FARMS AND ITS APPLICATION IN LINK SPAM DETECTION

by

Bin Zhou
B.Sc., Fudan University, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Spring 2007

APPROVAL

Name: Bin Zhou

Degree: Master of Science

Title of thesis: Mining Page Farms and Its Application in Link Spam Detection

Examining Committee: Dr. Jiangchuan Liu
Chair

Dr. Jian Pei, Senior Supervisor

Dr. Joseph Peters, Supervisor

Dr. Martin Ester, SFU Examiner

Date Approved: March 5th, 2007

Abstract

Understanding the general relations of Web pages and their environments is important with a few interesting applications such as Web spam detection. In this thesis, we study the novel problem of page farm mining and its application in link spam detection. A page farm is the set of Web pages contributing to (a major portion of) the PageRank score of a target page. We show that extracting page farms is computationally expensive, and propose heuristic methods. We propose the concept of link spamicity based on page farms to evaluate the degree of a Web page being link spam. Using a real sample of more than 3 million Web pages, we analyze the statistics of page farms. We examine the effectiveness of our spamicity-based link spam detection methods using a newly available real data set of spam pages. The empirical study results strongly indicate that our methods are effective.

Keywords:

page farm; link spam; PageRank; link spamicity

Subject Terms:

Web search engines; Text processing (Computer science); World Wide Web

To my parents,
and my sister.

"I like the dreams of the future better than the history of the past."

— THOMAS JEFFERSON*(1743 – 1826)*

Acknowledgments

My foremost thank goes to my supervisor and mentor Dr. Jian Pei. I thank him for his patience and encouragement that carried me on through difficult times, and for his insights and suggestions that helped to shape my research skills. His valuable feedback contributed greatly to this thesis. As an advisor, he taught me practices and skills that I will use in my future career.

I am grateful to my supervisor, Dr. Joseph Peters, for providing insightful comments and helpful suggestions that helped me to improve the quality of the thesis. His visionary thoughts and energetic working style have influenced me greatly. I also thank Dr. Martin Ester and Dr. Jiangchuan Liu for serving on my examining committee. I thank them for advising me and helping me in various aspects of my research, and their precious time reviewing my work.

My deepest thanks to Dr. Xiaoling Wang, Dr. Shuigeng Zhou and Dr. Aoying Zhou at Fudan University for inspiring me to work in the field of data mining, and educating and training me for my research skills.

I would also like to thank many people in our department, support staff and faculty, for always being helpful over the years. I thank my friends at Simon Fraser University for their help. A particular acknowledgement goes to Xu Cheng, Daniel Doucette, Zengjian Hu, Ming Hua, Dylan Huang, Wen Jin, Mag Lau, Flavia Moster, Rong She, Xiaojie Shen, Dan Wang, Feng Wang, Wendy Wang, Yang Wang, Arber Xu, Ji Xu, Xinghuo Zeng. I also would like to express my sincere thanks to my dear friends Sisi Huang, Edith Ngai, Wei Shen, Hao Shi, Tukang Song, Cheng Wang, Zhibin Wang, Raymond Wong, Tianyi Wu, Jian Xu, Yang Yang, Yao Yao, Xiaofeng Yuan, Hui Zhang, Jing Zhang, Ming Zhang, Junjie Zhou.

Last but not least, I thank my grandmother, my parents and my sister for always being

there when I needed them most, and for supporting me through all these years. I hope I will make them proud of my achievements, as I am proud of them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The World Wide Web (or simply the "Web") overwhelms us with immense amounts of widely distributed, interconnected, rich, and dynamic hypertext information. It has profoundly influenced many aspects of our lives by changing the ways we communicate, conduct business, shop, entertain, and so on. A recent study [33] which used Web searches in 75 different languages to sample the Web determined that there were over 11.5 billion Web pages in the publicly indexable Web as of January 2005. If including the pages that are not indexed by common Web search engines, for example, the dynamic pages generated by search queries, there were more than 550 billion pages on the Web [10]. In such an extremely large information collection, *Web search engines*, which are designed to help users to look for useful information on the Web, are absolutely necessary.

However, the abundant information on the Web is not stored in any systematically structured way. Such a situation poses great challenges to people who are seeking to effectively search for high quality information and to uncover the knowledge buried in billions of Web pages. In recent years, improving the quality of search results has become the main objective for Web search engines. However, Web search engines are now facing a number of challenging problems in maintaining or enhancing the quality of their performance [43], such as Web spam, content quality, quality evaluation, and so on. In general, most of these critical problems are related to one fundamental problem, *how to effectively and efficiently rank the pages on the Web.* Thus, to further understand the rankings of Web pages in detail is an interesting and important research problem in the Web mining area.

1.1 Motivation

Search engines have their roots in information retrieval systems. The first generation of search engines has a keyword index for the given corpus and responds to a keyword query with a ranked list of Web pages according to the keyword frequencies. A page containing one keyword more times is ranked higher than a page containing the same keyword less times. However, search results of keyword queries returned in this way may not be precise, in the sense that not all the pages matching a set of keywords in the query can be ranked nicely to reflect their relativities. A better bet is to rate each Web page by evaluating how likely it satisfies the user's information need by considering the whole content of the page, sort them in the descending order of this likelihood, and present the results in a ranked list. This is how the content-based Web search engines work.

Since only a part of the user's information need is expressed through keyword queries, there can be no algorithmic way of ensuring that the ranking strategies always favor the information need. As the Web is growing faster and faster, the content-based ranking strategies often cannot meet the high quality expectations from users. Thus, finding good methods using not only the content information, but also some other features to rank Web pages effectively and efficiently becomes an essential task in Web search. Many studies have been dedicated to effective ranking methods. In recent years, due to the great success of Google, the link structure-based ranking strategies, such as HITS [48] and PageRank [59], have shown the high effectiveness to rank Web pages. The link structure-based ranking methods try to remedy the imprecise problem inherent in content-based ranking methods by supplementing precision with notions related to "prestige" that is independent of any information need or query. Roughly speaking, the prestige of a page is proportional to the sum of the prestige scores of pages linking to it. The prestige scores can reveal the importance of Web pages very well. As a result, most of the popular search engines currently adopt some link structure-based ranking methods, among which Google's PageRank is the most representative one.

On the other hand, driven by the huge potential benefit of promoting rankings of Web pages, many dirty tricks have been attempted to boost page rankings by making up some artificially designed link structures, which is known as *link spam* [6, 13, 18, 25, 34, 35, 36, 43, 51]. The term "spam" here refers to any deliberate human action that is meant to trigger an unjustifiably favorable relevance or importance for some Web pages comparing to the true value of the page.

Generally, the link structure-based ranking score of a Web page comes from some other pages linking to it. In order to fully understand the link structure-based ranking, a question essential but largely and generally remaining open is as follows.

> *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 a few interesting applications such as link spam detection and Web community (typically, collections of Web pages that share some common interests in a specific topic) identification and analysis. For example, we may detect link spam pages effectively if we can understand the "normal" ways that Web pages collect their ranking scores. A Web page is a suspect of link spam if the way it collects its ranking score is substantially different from those normal models.

This thesis tries to make good progress in answering the proposed question. Moreover, we investigate the application of our proposed model in link spam detection.

1.2 Contributions

In this thesis, we propose the novel concept of *page farm* and study the problem of *page farm mining*, and illustrate its application in link spam detection. In particular, the general ideas and our major contributions are listed as follows.

- First, *we study 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 propose 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 study the computational complexity of finding page farms, and show that it is NP-hard. Then, we develop a practically feasible greedy method to extract approximate page farms.

- Second, *we empirically analyze statistics of page farms using over* 3 *million Web pages randomly sampled from the Web.* We have a few interesting and exciting findings. Most importantly, the landscapes of page farms tend to follow the power law distribution. Moreover, the landscapes of page farms strongly reflect the importance of the Web pages, and their locations in their Web sites. To the best of our knowledge, this is the first empirical study on extracting and analyzing page farms.

- Third, *we investigate the application of page farms in link spam detection.* We propose two methods. First, we measure the utility of a page farm, that is, the "perfectness" of a page farm in obtaining the maximum PageRank score, and use the utility as an index of the likeliness of link spam. Second, we use the statistics of page farms as the indicator of the likeliness of link spam. Using those measures we can detect link spam pages.
- Last, *we evaluate our link spam detection methods using a newly available real data set.* The pages are labeled by human experts. The experimental results show that our methods are effective in detecting spam pages.

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows:

- In Chapter 2, we present an overview of the related work systematically.
- In Chapter 3, a novel Web link structure model, page farm model, is introduced. We study the computational complexity of extracting page farms, and its NP-hard property is verified by complete and systematic theoretical analysis.
- We develop the methods for extracting page farms in Chapter 4, and then report some empirical results for the landscapes of page farms. The experimental evaluations reveal some interesting and exciting findings.
- We investigate link spam detection using page farms in Chapter 5, and report an empirical evaluation on a newly released spam test collection data set. The experimental results strongly show the effectiveness of our methods.
- The thesis is concluded in Chapter 6. We summarize the major characteristics of the page farm model, and discuss some interesting extensions and applications, then present some future directions.

Chapter 2

Related Work

In this chapter, we give an overview of the related work systematically. In general, our study is highly related to the previous work in the following four aspects: (1) social network analysis; (2) Web link structure modeling and analysis; (3) Link Structure-based Ranking and Web Communities; and (4) Web spam detection.

2.1 Social Network Analysis

Social network analysis has been studied extensively and substantially (for example, see [63, 60] as two textbooks). Here, we present some essential and important work.

A social network is a social structure made of nodes which are generally individuals or organizations. A social network indicates the ways in which the nodes in the network are connected through various social familiarities ranging from casual acquaintance to close familial bonds. The term was first coined by J. A. Barnes [8] in 1954. A social network can be modeled as a graph, where the nodes in the network are the individuals or organizations, while the links show relationships and interactions between the nodes. Social network plays a fundamental role as a medium for the spread of information, ideas, and influence among its members.

Social network analysis [53, 15, 30, 64, 40, 47, 65] is the mapping and measuring of relationships and interactions between the nodes in the network. Social network analysis has emerged as a key technique in modern sociology, anthropology, sociolinguistics, geography, social psychology, information science and organizational studies.

Social network analysis consists of a set of methods for analyzing social structures. The methods are specifically geared towards an investigation of the relational aspects of the structures. The use of these methods, therefore, depends on the availability of relationship data rather than attribute data. To understand social networks and their participants, people evaluate the locations of the participants in the network. Measuring the location in the network is to find the centrality of a node such as degree centrality, betweenness centrality and closeness centrality which will be illustrated in the next paragraph. These measures give us insight into the various roles and groupings in a network – who are the connectors (that is, the nodes connecting two groups), leaders (that is, the most important nodes in the group), isolates (that is, the nodes not participating in some other groups)? Where are the clusters (that is, a group of nodes that are similar to one another within the same cluster and are dissimilar to the nodes in other clusters) and who are in them? Who are in the core (that is, a group of nodes that are most important and representative in the social network) of the network? And who are on the periphery (that is, a group of nodes that are least important in the social network)? For example, a friendship network in the real life effectively shows the distinction between the three most popular individual centrality measures: Degree Centrality, Betweenness Centrality, and Closeness Centrality [63, 60].

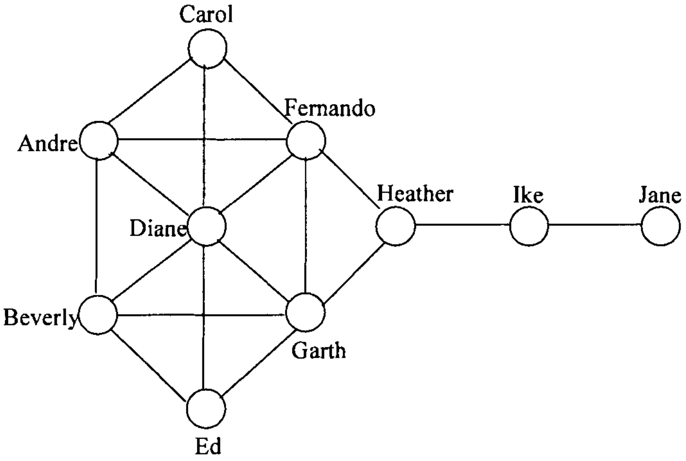


Figure 2.1: An example of a friendship network in the real life.

Figure 2.1 gives an concrete example of "Kite's Network" [63], which is developed by D. Krackhardt, a leading researcher in the area of social network analysis. Two nodes are connected if they regularly interact with each other in some way. Social network researchers measure the network activity for a node by using the concept of degrees which is the number of direct connections a node has. This is defined as the *Degree Centrality.* In the Kite's network shown in Figure 2.1, Diane has the most direct connections in the network, making her the most active node in the network. The *Betweenness Centrality* is used to measure the importance of a node in the network. For example, in Kite's Network, while Diane has many direct ties, Heather only has a few direct connections which is fewer than the average in the network. However, Heather has one of the best locations in the network, that is, she is between two important subnetworks. The good news is that she plays a powerful role in the network, but the bad news is that she is a single point of failure. Without her, Ike and Jane would be disconnected from the network. A node with high betweenness has great influence over what flows in the network. In addition, Fernando and Garth have fewer connections than Diane, yet the patterns of their direct and indirect ties allow them to access all the other nodes in the network more quickly than anyone else. They have the shortest paths to all the others. In other words, they are close to everyone else. They are in an excellent position to monitor the information flow in the network. This is so called the *Closeness Centrality.*

One of the challenging problems in social network analysis is the social community identification problem. A social community is a group of social members sharing some common interests. The process by which communities come together, attract new members, and develop over time is a central research issue in the social sciences. The political movements, professional organizations, and religious denominations all provide fundamental examples of such communities. The tendency of people to come together and form groups is inherent in the structure of society; and the way in which such groups take shape and evolve over time is a theme that runs through large parts of social science research [21]. In general, the social community identification problem corresponds to the graph partition problem and the node clustering problem. There is a large body of work on identifying tightly connected clusters within a given graph (for example, see [29, 27, 32, 44, 56]).

Some previous work is focusing on analyzing the evolution of social networks. Research over the past a few years has identified classes of properties that many real world networks obey. One of the main areas of focus has been on degree power laws, showing that the set of node degrees has a heavy tailed distribution. Such degree distributions have been identified in various social networks [17]. Some Other properties of the evolution include the "small-world phenomenon" [63], popularly known as "six degrees of separation", which states that real large social network graphs have surprisingly small average diameters.

How Is Our Study Different?

The 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 studies on the concept of "ecology" environments of Web pages, which we refer to page farms in this thesis. We are also the first one to analyze the distributions of page farms.

2.2 Web Link Structure Modeling and Analysis

The link structure of the Web can be viewed as a directed graph in which each vertex is a Web page, and each edge is a hyperlink between two pages. The Web graph has some interesting properties, such as the power law degree distribution [50, 49, 14] and a small average diameter [2]. A number of stochastic models for the Web graph have been proposed to better understand and predict the statistical properties of the Web.

A popular Web graph model is the preferential attachment model [7]. We refer to this model and its variants as random Web graph models. In this model, vertices and edges are dynamically added to the graph, such that the probability that an existing vertex gets a new link depends positively on its current degree. The resulting process generates graphs whose degree distributions follow the power law distribution [50, 49, 14].

Some previous work is focusing on analyzing the link structure of the Web graph. In [2], Albert et al. reported an intriguing finding that most pairs of pages on the Web are separated by only a few hyperlinks. The lengths of the link paths (that is, the number of hyperlinks) are under 20 in most cases. They also predicted that this number would grow logarithmically with the size of the Web. This is viewed as a "small world" phenomenon on the Web. At the same time, Broder et al. [14] conducted systemical experiments on a large sampled Web data set. The experimental results reveal an even more detailed and subtle picture: "most ordered pairs of pages cannot be bridged at all and there are significant numbers of pairs that can be bridged, but only using paths going through hundreds of intermediate pages". As a conclusion, the connectivity of the Web is strongly limited by a high-level global structure.

Meanwhile, Broder et al. in [14] showed that the macroscopic structure of the Web has a Bow-Tie shape composed of 5 main regions as shown in Figure 2.2. Most of the pages in the

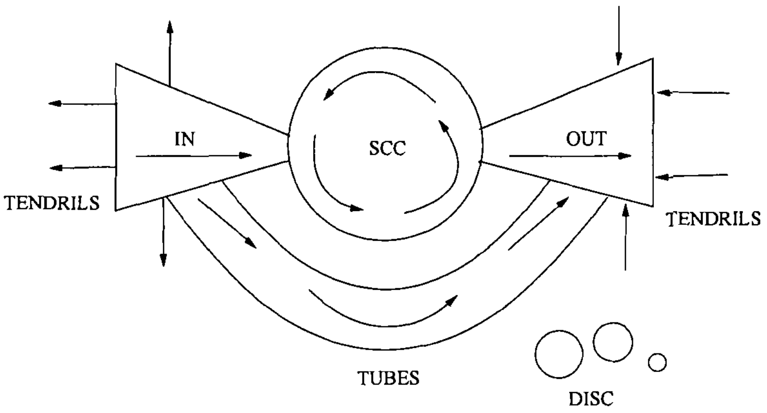


Figure 2.2: Connectivity of the Web: the Bow-Tie structure.

Web graph form a single connected component. This connected component can be further divided naturally into four pieces. The first piece is called "SCC" (Strongly Connected Component), which contains the pages that can reach one another along directed links. The second and the third pieces are called "IN" and "OUT". The component "IN" contains the pages with links that lead to "SCC", but not vice versa. For example, new sites that people have not yet discovered and linked to are in this component. The component "OUT" contains the pages that are reachable from "SCC", but not vice versa, such as corporate Web sites that contain only internal links. The fourth piece, "TUBES and TEDNRILS", contains all the other pages that are reachable from "IN" or lead to "OUT". The remaining region that is not a part of the connected component is referred to as the last component "DISC", which contains all the other disconnected components in the Web graph. Moreover, based on the large sample Web data set they crawled, Broder et al. found a surprising fact that the size of the component "SCC" is relatively small. Actually, the experimental results showed that all the four components, "SCC", "IN", "OUT", and "TUBES and TENDRILS", have roughly the same size.

Recently, some models in graph theory, such as the neighborhood graph theory, have been introduced to analyzing the Web graph. Neighborhood graphs have been studied for a long time [23]. Given a graph $G = (V, E)$ and a node $p \in V$, a simple neighborhood graph for p is a subgraph of G which only contains the nodes that have an edge pointing to p. In the Web graph, given a page p, Nargis et al. [55] introduced the concept of the neighborhood graph of p as a subgraph of the original Web graph which contains the pages that are at the distance less than a given distance threshold k. Analyzing such kind of neighborhood graphs has some useful applications, such as to find communities of related Web pages within a specific distance of a given Web page, and to understand the structural and statistical properties of the local structures of the Web graph.

How Is Our Study Different?

The previous work of neighborhood graph-based Web structure modeling simply considers the near neighbor pages for a given target page. In order words, the distance to the target page is the only factor considered. To the best of our knowledge, our page farm model is the first one to introduce the contributions of ranking scores as the weights for the neighbor pages. The pages in the page farm have high contributions to the target page. The page farm may have good potential to reflect that the pages in the farm have tight relationships with the target page.

2.3 Link Structure-based Ranking and Web Communities

A few link structure-based ranking methods, such as PageRank [59] and HITS [48], were proposed to assign scores to Web pages to reflect their importance. Both HITS and PageRank try to remedy the imprecise problem inherent in keyword queries by supplementing precision with notions related to "prestige" in social network analysis. In PageRank, each page on the Web has a measure of prestige that is independent of any information need or query. Roughly speaking, the prestige of a page is proportional to the sum of the prestige scores of pages linking to it. In HITS, a query is used to select a subgraph from the Web. From this subgraph, two kinds of nodes are identified: authoritative pages to which many pages link, and hub pages that contain comprehensive collections of links to authoritative pages.

Although there are technical differences between PageRank and HITS, the two measures are defined recursively: the prestige of a node in PageRank depends on the prestige of other nodes. In HITS, the measure of being a good hub page depends on how good the neighbor pages are as authoritative pages, and similar for the measure of authoritative pages. Both PageRank and HITS involve computing eigenvectors for the adjacency matrix, or a matrix derived from the Web or a suitable relevant subgraph of the Web.

Using link structure-based analysis, the previous studies have developed various methods to identify Web communities, that is, collections of Web pages that share some common interest in a specific topic. Link structure-based Web community identification is different from the work in social community identification described in Section 2.1 in that the link structure-based ranking algorithms are used to help to solve the problem.

For example, Gibson et al. [31] developed a notion of hyper-linked communities on the Web through an analysis of the link topology. As another example, Kleinberg [48] showed that the HITS algorithm, which is strongly related to spectral graph partitioning, can identify hub and authoritative pages. Hubs and authorities are especially useful for identifying key pages related to some communities. However, using HITS to enumerate all members of a community sometimes might be problematic because the communities in which one is interested may be overshadowed by a more dominant community.

To tackle the problem, Flake et al. [28] modeled a Web community as a collection of Web pages in which each member page has more hyperlinks within the community than outside the community. However, finding optimal communities in this definition is NP-hard because it essentially belongs to the family of graph partitioning problems. Flake [28, 29] showed that the Web community problem can be recast into a maximum flow framework to analyze the flows between graph vertices.

In general, link structure analysis seldom considers the text information of Web pages. Bharat [11] indicated that, without auxiliary text information, both PageRank and HITS have only limited success in identifying Web communities.

How Is Our Study Different?

The Web community identification problem is to find a set of pages that share the same interest. In other words, it is a global view of pages on the Web. The previous link structure-based ranking methods and their applications do not analyze the environments of Web pages. Our page farm model is the first one trying to fill up this gap between the global views and the individual Web pages.

2.4 Web Spam Detection

Most of the popular Web search engines currently adopt some link structure-based ranking algorithms, such as PageRank and HITS. Driven by the huge potential benefit of promoting rankings of pages, many attempts have been conducted to boost page rankings by making up some linkage structures, which is known as link spam [6, 13, 18, 25, 34, 35, 36, 43, 51]. The term "spam" here refers to any deliberate human action that is meant to trigger an unjustifiably favorable relevance or importance for some Web pages comparing to the page's true value.

Because PageRank scores are determined based on the link structures of the Web, PageRank is a natural target of link spam. Gyöngyi et al. [35, 36] referred link spam to the cases where spammers set up structures of interconnected pages, called link spam farms, in order to boost the link structure-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 ranking of the target page, and some external links accumulated from pages outside the spam farm. Based on this model, given a fixed number of boosting pages, the optimal link structure by which the target page can achieve the highest PageRank score is addressed in [35]. Moreover, Gyöngyi et al. [35] also showed the link spam alliance (that is, the collaboration of spammers) and the corresponding optimal link structures.

Some methods have been proposed to detect link spam. Fetterly et al. [25] adopted statistical analysis to detect link spam. Several distribution graphs, such as the distribution of in-degrees and out-degrees, were drawn. Most of these distributions were modeled well by some form of power law distribution. The outliers in the results were marked as spam candidates. By manually checking these candidates, a majority of them were found to be spam. Wu and Davison [66] proposed an algorithm for link spam detection. It first generated a seed set of possible spam farm pages based on the common link set between incoming and outgoing links of Web pages. Then, spam pages were identified by expanding the seed set. Recently, Gyöngyi et al. [34] introduced the concept of spam mass, a measure of the impact of link spam on a page's ranking. [34] discussed how to estimate spam mass and how the estimations can help to identify pages that benefit significantly from link spam.

Some other link spam detection methods resemble PageRank computation. Benczur et al. [9] proposed a method called SpamRank, which was based on the concept of personalized PageRank that detected pages with an undeserved high PageRank score. They defined SpamRank by penalizing pages that originate a suspicious PageRank share and personalizing PageRank on the penalties. Gyöngyi et al. [37] described an algorithm, called TrustRank, to combat Web spam. The basic assumption of TrustRank was 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. They then followed an approach similar to PageRank: the trust score was propagated via out-links to other Web pages. Finally, after convergence, the pages with high trust scores were believed to be good pages. However, TrustRank was vulnerable in the sense that the seed set used by TrustRank may not be sufficiently representative to cover well the different topics on the Web. Also, for a given seed set, TrustRank had a bias towards larger communities. To address the above issues, Wu et al. [67] proposed the use of topical information to partition the seed set and calculate the trust scores for each topic separately. A combination of these trust scores for a page was 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 proposed so far adopted statistical analysis. For example, in [25], Fetterly et al. 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, in [26], Fetterly et al. 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, and presented methods for detecting such pages by identifying popular shingles. Recently, Ntoulas et al. [58] presented a number of heuristic methods for detecting content-based spam that essentially extend the previous work [25, 26]. Some of those methods were more effective than the others, however, the methods may not identify all of the spam pages when used in isolation. Thus, [58] combined the spam detection methods to create a highly accurate C4.5 classifier [39] to detect term spam pages.

How Is Our Study Different?

In some link spam detection methods [9, 34, 37, 58, 67, 69], the concept of link spam farm is used to conceptually capture the set of Web pages that achieve the link spam. However, there is neither method proposed to extract link spam farms from the Web nor empirical studies on the link spam farms.

We use the page farms and particularly the utility and the characteristics of the page farms to detect link spam 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.

Chapter 3

Page Farms

In this chapter, we propose our novel Web link structure model, page farm model. We study the computational complexity of extracting page farms, and its NP-hard property is verified by complete and systematic theoretical analysis.

3.1 Page Farm Model

The Web 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 an edge $p \longrightarrow q$. An edge $p \longrightarrow q$ 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 $p \longrightarrow q$ is formed. Such structure modeling not only can make the Web graph simple, but also can let some mathematical models, such as the Markov chain model [3], be suitable for analyzing the Web graph. Hereafter, by default our discussion is about a directed Web graph $G = (V, E)$.

PageRank [59] 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 [59] is defined as

$$PR(p, G) = d \sum_{p_i \in M(p)} \frac{PR(p_i, G)}{OutDeg(p_i)} + \frac{1-d}{N}, \tag{3.1}$$

where $M(p) = \{q | q \longrightarrow p \in E\}$ is the set of pages having a hyperlink pointing to p, $OutDeg(p_i)$ is the out-degree 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 the random transitions on the Web and a typically used value is 0.85, and $N = |V|$ is the total number of pages in the Web graph. The second additive term on the right side of the equation, $\frac{1-d}{N}$, is traditionally referred to as the random jump probability and corresponds to a minimal amount of PageRank score that every page gets by default.

To calculate the PageRank scores for all pages in a graph, one can assign a random PageRank score value to each node in the graph, and then apply Equation 3.1 iteratively until the PageRank scores in the graph converge. As proved in [13, 51], given a Web graph, no matter what kind of orders the pages being considered for PageRank computation, the PageRank vector will be converged after several iterations. Even more, the PageRank vector is stochastic. Thus, the converged PageRank score for a given page is unique.

For a Web page p, can we analyze what other pages contribute to the PageRank score of p? An intuitive 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.

Generally, 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.* According to Equation 3.1, 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 the previous studies [2, 14], the major part of the Web is strongly connected. Albert et al. [2] indicated that the average distance of the Web is 19. 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 is reported in [14]. This result is based on a large sample of Web data set the authors crawled. 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 3.1, PageRank contributions are only made by the out-edges. 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 3.1. Suppose we want to void page v in the graph for PageRank calculation. Removing v from the graph also reduces the out-degree of u, and thus change the PageRank contribution from u to p. Moreover, simply removing v alters the random jump probability into each page which is undesirable. Instead, we should retain v but remove the out-link $v \rightarrow p$.

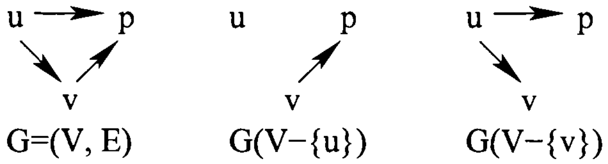


Figure 3.1: Voiding pages and induced subgraphs.

For a set of vertices U, the *induced subgraph* 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 3.1 shows two examples.

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 given by

$$Cont(U, p) = \frac{PR(p, G(U))}{PR(p, G)} \times 100\%. \tag{3.2}$$

PageRank contribution has the following property, which follows with Corollary 3 to be discussed in Section 4.1.

Corollary 1 (Monotonic contributions) *Let p be a page and U, W be two sets of pages such that $U \subseteq W$. Then, $0 \leq Cont(U, p) \leq Cont(W, p) \leq 1$.*

Proof. This corollary can be proved based on page contribution and path contribution which are defined in Chapter 4. ■

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

Definition 1 (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. ■

3.2 Complexity of Page Farm Extraction

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

Theorem 1 (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 [46], 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 $G = (V, E)$. The vertices and the edges are created in three steps. First, vertex $v_0 \in V$ is created as a "knapsack". Second, for each item u_i, we create a vertex v_i. 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 $v_{i,1}, \cdots, v_{i,w(u_i)-1}$, are inserted as a path $v_i \rightarrow v_{i,1} \rightarrow v_{i,w(v_i)-1} \rightarrow v_0$.

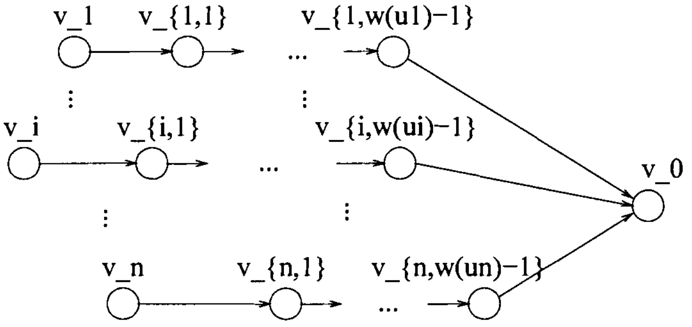


Figure 3.2: Reducing the knapsack problem for θ-farm extraction problem.

Figure 3.2 illustrates the construction of the graph G. As the result, in G, 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 G 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 3.1 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 $v_{i,1}, \cdots, v_{i,w(u_i)-1}$ 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 $G' \subseteq G$ 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 G' is $val(u_i)$.

Last, in graph $H \subseteq G$ which contains directed paths from $v_{j_1}, \cdots, v_{j_l}$ to v_0 $(1 \leq j_1, \cdots, j_l \leq n)$, 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 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. 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 (that is, the 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 ((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. ■

We notice that finding the exact (θ, k)-farms is also computationally expensive on large networks.

Corollary 2 ((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. The proof of the NP-hardness for finding a (θ, k)-farm is similar to that for finding a θ-farm. We construct a reduction from the knapsack problem.

We do the transformation in the same way as shown in the proof of Theorem 1. We set the distance parameter $k = max(w(u_i))$ where $1 \leq i \leq n$. Thus, in Figure 3.2, all the nodes are at the distance at most k to the node v_0. The transformation can be done in polynomial time.

By the above transformation, 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 $(\frac{K}{PR(v_0,G)}, k)$-farm of v_0 of size at most W. ■

Typically, link spam is a local activity. Comparing to θ-farm, (θ, k)-farm even reduces the noisy information and can capture those most important and nearest contributors better. Thus, in the later analysis, our discussions focus on (θ, k)-farms by default.

Based on Definition 2, a page farm U is a set of pages. We can easily obtain an induced graph $G(U)$ by adding the links between pages in the farm. In some applications, people are interested in not only the pages, but also the link structures among those pages. Hereafter, if there is no confusion, we may also refer the page farms to an induced subgraph $G(U)$ of the whole Web graph, where the nodes are those pages in U and the edges are induced by the pages in U.

Chapter 4

Page Farm Analysis

In this chapter, we develop the methods for extracting page farms, and then report some empirical analysis for the landscapes of page farms. The experimental results reveal some interesting and exciting findings, and show some great potentials of the proposed page farm model.

4.1 Extracting Page Farms

In this section, we first give a simple greedy method to extract page farms, and analyze its inefficiency. Then, we propose a practically feasible method to extract approximate page farms.

4.1.1 A Simple Greedy Method

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

Definition 3 (Page contribution) For a target page $p \in V$, the **page contribution** of page $v \in V$ to the PageRank score of p is

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

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

Definition 3 is based on intuitive observation, and it is reasonable and easy to understand. If $v = p$, according to the original PageRank formula in Equation 3.1, $\frac{1-d}{N}$ 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 $PCont(v,p) = PR(p,G) - PR(p,G(V-\{v\}))$.

Example 1 (Page contribution) Consider the simple Web graph G in Figure 3.1. The induced subgraphs $G(V-\{u\})$ and $G(V-\{v\})$ are also shown in the figure.

Let us consider page p as the target page, and calculate the page contributions of the other pages to the PageRank of p. Notice that $N = 3$. According to Equation 3.1, 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

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

■

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 $u_1, u_2, \cdots$. Then, we select the top-l

pages $u_1, \cdots, u_l$ as an approximation of the θ-farm of p such that $\frac{PR(p,G(V-\{u_1,\cdots,u_l\}))}{PR(p,G)} \geq \theta$ and $\frac{PR(p,G(V-\{u_1,\cdots,u_{l-1}\}))}{PR(p,G)} < \theta$.

The above greedy method is simple. Since the algorithm greedily selects the pages with the highest page contribution to the target page, the results can capture the environment of the target page nicely. Unfortunately, it is inefficient for large Web graphs. First, it assumes that the whole Web graph is available, which may not be true for many situations. Second, in order to extract the page farm for a target page p, we have to compute the PageRank of p in 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 amount of Web pages and hyperlinks.

4.1.2 Path Contributions

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

Definition 4 (Path contribution) Consider Web graph $G = (V, E)$ and target page $p \in V$. Let $P = v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_n \rightarrow p$ 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

$$LCont(P,p) = \frac{1}{N} d^{n+1}(1-d) \prod_{i=0}^{n} \frac{1}{OutDeg(v_i)}, \tag{4.1}$$

where $OutDeg(v_i)$ is the out-degree of page v_i, and N is the total number of pages in the Web graph. ■

The PageRank scores can be calculated using the path contributions.

Theorem 2 (Path contribution) *The PageRank of p is*

$$PR(p,G) = \frac{1-d}{N} + \sum_{v \in W(p)} \left(\sum_{P \in DP(v,p)} LCont(P,p) \right), \tag{4.2}$$

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

Proof. Recall the original PageRank formula in Equation 3.1

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

where $M(p)$ is the set of pages having a hyperlink pointing to p, $OutDeg(p_i)$ is the out-degree of p_i, d is a damping factor, and N is the total number of pages in the Web graph.

Suppose $|M(p)| = m$, that is, there are m pages having a hyperlink pointing to p. Without loss of generality, we use $p_1, p_2, \cdots, p_m$ to denote those m pages. Thus, we have

$$PR(p, G) = d \sum_{i=1}^{m} \frac{PR(p_i, G)}{OutDeg(p_i)} + \frac{1-d}{N}. \tag{4.3}$$

As shown in [13, 51], the PageRank formula is based on the Markov chain model [3], which implies that no matter which order is used to compute the PageRank score vector based on Equation 4.3, after a few steps, finally it will converge to the stationary PageRank vector. Thus, given the Web graph, the converged PageRank score of p is unique. So if we can show that $PR(p, G) = \frac{1-d}{N} + \sum_{v \in W(p)} \sum_{P \in DP(v,p)} LCont(P, p)$ is actually a solution to Equation 4.3, we show the correctness of Theorem 2.

We replace $PR(p, G)$ and $PR(p_i, G)$ in Equation 4.3 by the following two formulae

$$\begin{cases} PR(p, G) = \frac{1-d}{N} + \sum_{v \in W(p)} \sum_{P \in DP(v,p)} LCont(P, p) \\ PR(p_i, G) = \frac{1-d}{N} + \sum_{v \in W(p_i)} \sum_{P \in DP(v,p_i)} LCont(P, p_i) \end{cases}$$

Then, we have

$$\sum_{v \in W(p)} \sum_{P \in DP(v,p)} LCont(P, p) = d \sum_{i=1}^{m} \frac{\frac{1-d}{N} + \sum_{v \in W(p_i)} \sum_{P \in DP(v,p_i)} LCont(P, p_i)}{OutDeg(p_i)}. \tag{4.4}$$

We then replace $LCont(P, p)$ and $LCont(P, p_i)$ using Definition 4. In order to make the formula simple and easy to understand, we introduce some notations first.

Given two pages p and v_1, suppose there is a link path P from v_1 to p, denoted as $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow p$, we use $|P|$ to denote the length (the number of edges) of the path P, that is, $|P| = n$. We use $\pi(P)$ to denote the contribution propagation, that is, $\pi(P) = \prod_{i=1}^{n} \frac{1}{OutDeg(v_i)}$. We use P_p to denote a link path pointing to page p. We also use $\mathcal{P}_p$ to represent the set of link paths pointing to page p.

The meaning of Theorem 2 is to find out all the different link paths pointing to p, and then sum up all the path contributions. Thus, Equation 4.4 can be rewritten as following

$$\sum_{P_p} \frac{1-d}{N} d^{|P_p|}\pi(P_p) = d\sum_{i=1}^{m} \frac{\frac{1-d}{N} + \sum_{P_{p_i}} \frac{1-d}{N} d^{|P_{p_i}|}\pi(P_{p_i})}{OutDeg(p_i)} \tag{4.5}$$

Now we prove Equation 4.5. Since P_p and P_{p_i} represent link paths pointing to page p and p_i, respectively. We also know that there is a hyperlink directly from p_i to p, thus one link path in the set of $\mathcal{P}_{p_i}$ actually corresponds to one link path in the set of $\mathcal{P}_p$. For example, given any link path $P_{p_i} : v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow p_i$, we can get a corresponding link path $P_p : v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow p_i \rightarrow p$. In other words, for any link path P_{p_i}, there is a link path P_p such that $|P_p| = |P_{p_i}| + 1$ and $\pi(P_p) = \frac{\pi(P_{p_i})}{OutDeg(p_i)}$.

The set of $\mathcal{P}_p$ can be divided into two subsets: one is the set of paths generated from $\mathcal{P}_{p_i}$, which is denoted by $\mathcal{P}_p^1$, and the other set contains the direct links from p_i to p, which is denoted by $\mathcal{P}_p^2$. Thus, the left handside of Equation 4.5 can be rewritten as

$$\begin{aligned}
\sum_{P_p} \frac{1-d}{N} d^{|P_p|}\pi(P_p) &= \sum_{P_p^1 \in \mathcal{P}_p^1} \frac{1-d}{N} d^{|P_p^1|}\pi(P_p^1) + \sum_{P_p^2 \in \mathcal{P}_p^2} \frac{1-d}{N} d^{|P_p^2|}\pi(P_p^2) \\
&= \sum_{i=1}^{m}\Big(\sum_{P_{p_i}} \frac{1-d}{N} d^{|P_{p_i}|+1} \frac{\pi(P_{p_i})}{OutDeg(p_i)}\Big) + \sum_{i=1}^{m} \frac{1-d}{N} \frac{d}{OutDeg(p_i)} \\
&= d\sum_{i=1}^{m} \frac{\frac{1-d}{N} + \sum_{P_{p_i}} \frac{1-d}{N} d^{|P_{p_i}|}\pi(P_{p_i})}{OutDeg(p_i)}
\end{aligned} \tag{4.6}$$

From Equation 4.6, we can conclude that

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

is actually a solution to Equation 4.3. Since the solution is unique, we have Theorem 2. ■

Moreover, page contributions can also be calculated using path contributions. Applying Theorem 2 to Definition 3, we have the following result.

Corollary 3 (Page and path contributions) *For vertices p and q in Web graph $G = (V, E)$, if the in-degree 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

$$PCont(q,p) = \sum\nolimits_{\text{path } P_1 \text{ from } q \text{ to } p} LCont(P_1,p) + \sum\nolimits_{v \in W_q(p)} \sum\nolimits_{\text{path } P_2 \text{ from } v \text{ to } p \text{ through } q} LCont(P_2,p),$$

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

Proof. According to Definition 3, for vertices p and q ($q \neq p$) in Web graph $G = (V, E)$,

$$PCont(q,p) = PR(p,G) - PR(p, G(V - \{q\})). \tag{4.7}$$

We apply Theorem 2 to Equation 4.7, and have the following

$$\begin{aligned} PCont(q,p) &= (\frac{1-d}{N} + \sum_{P \in G(V)} LCont(P,p)) - (\frac{1-d}{N} + \sum_{P' \in G(V-\{q\})} LCont(P',p)) \\ &= \sum_{P \in G(V)} LCont(P,p) - \sum_{P' \in G(V-\{q\})} LCont(P',p) \end{aligned} \tag{4.8}$$

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 4.8, we have Corollary 3. ■

The monotonic contribution property of PageRank contributions (Corollary 1) can be derived from Corollary 3.

Corollary 1 (Monotonic contributions) Let p be a page and U, W be two sets of pages such that $U \subseteq W$. Then, $0 \leq Cont(U,p) \leq Cont(W,p) \leq 1$.

Proof. A path contribution is non-negative. When some vertices are voided for PageRank calculation, some paths are destroyed. Thus, the PageRank score 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 3.1 again. There are three paths to the 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

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

Using Corollary 3, we have

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

Moreover, by Theorem 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. ■

Comparing to page contributions, path contributions are cheaper to be computed. We can derive them directly from the graph structure using the out-degrees of pages. We even do not need the PageRank scores of pages. On the other hand, recall that computing a page contribution directly using Definition 3 has to iteratively compute the PageRank score of the target page in an induced subgraph until the score converges.

Interestingly, simultaneously to our study, Gyöngyi et al. proposed a measure on PageRank contribution from Web pages and link paths [34]. 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).

To illustrate the difference, consider the directed graph in Figure 4.1. In their definition, pages p_1 and p_2 have the same contribution to page p. In our definition, page p_2 contributes

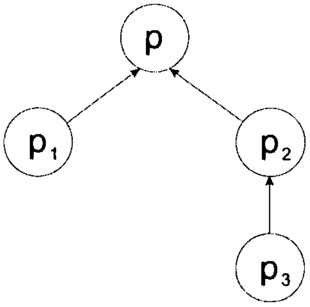


Figure 4.1: An example showing the difference between two measures of PageRank contribution.

more. Our argument is that p_2 has a higher PageRank score due to its receiving a link from p_3, and in sequel contributes more to the PageRank score of p.

We also notice that, simultaneously to our study, a similar idea called branching contribution is presented in [68]. Basically, the general idea of branching contribution is same to path contribution.

4.1.3 Extracting (θ, k)-farms

Can we approximate (θ, k)-farms of Web pages efficiently using path contributions developed in Section 4.1.2?

We propose a greedy algorithm in Figure 4.2. The algorithm takes the immediate neighbors of p (that is, those pages having 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 greedy algorithm, only those pages whose distances to p are no more than k may be searched. Moreover, each of such pages can be included into the candidate set at most once. Therefore, it is much more efficient than the simple greedy algorithm given in Section 4.1.1.

```
Input: a Web graph $G = (V, E)$, a target page $p \in V$,
    a damping factor $d$, parameters $\theta$ and $k$;
Output: an approximate $(\theta, k)$-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
    $q = \arg\max_{q \in S}\{PCont(q, p)\}$;
    $Farm = Farm \cup \{q\}$;
    $S = S - \{q\} \cup \{q' | q' \rightarrow q \in E \wedge dist(q', p) \leq k\}$;
}
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 Theorem 2 and Corollary 3;
return $\frac{PR(p,G(Farm \cup \{p\}))}{PR(p,G)}$;
```

Figure 4.2: A greedy algorithm to extract an approximate (θ, k)-farm of a target page.

Moreover, Theorem 2 and Corollary 3 help to compute the PageRank contribution efficiently. 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. Second, once a page is added into the page farm, the contributions of the pages in the candidate set can be updated accordingly.

4.2 The Landscapes of Page Farms – An Empirical Analysis

In this section, we report an empirical analysis of page farms of a large sample from the Web. To test the page farm extraction from the Web, we used a data set generated by the Web crawler "WebVac" from the Stanford WebBase project[1]. Some prior studies [41, 42, 45, 67]

[1] http://www-diglib.stanford.edu/~testbed/doc2/WebBase

used the same data set in their experiments. The Web crawler randomly crawls up to a depth of 10 levels and fetches a maximum of 10 thousand pages per site. The whole directed Web graph file as of May, 2006 is about 499 GB and contains about 93 million pages.

Limited by the computational resource available to us, in our experiments, we only used a random sample subgraph of the whole Web graph. The **Web page sample set** we used contains 3,295,807 pages from more thank 60,000 sites, and is about 16 GB. Each page in our data set has a viable URL string.

All the experiments were conducted on a PC computer running the Microsoft Windows XP SP2 Professional Edition operating system, with a 3.0 GHz Pentium 4 CPU, 1.0 GB main memory, and a 160 GB hard disk. The program was implemented in C/C++ using Microsoft Visual Studio. NET 2003.

4.2.1 Extracting Page Farms

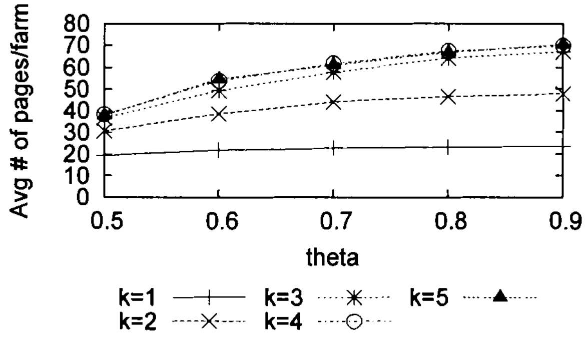


Figure 4.3: The effects of parameters θ and k on the page farm extraction (**Web page sample set**).

To understand the effects of the two parameters θ and k on the page farms extracted, we extracted the (θ, k)-farms using different values of θ and k, and measured the average size of the extracted farms. Figure 4.3 shows the results on the whole **Web page sample set** of over 3 million pages.

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. In this case, the farms the algorithm extracted are not exactly (θ, k)-farms but only k-neighbor pages. Therefore, when k increases, the average page farm size increases. However, when k is 3 or larger, the page farm size is stable. This verifies our observation that the near neighbor pages contribute more than the remote ones, and the PageRank score of a page is mainly determined by its near neighbors.

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. The larger and the denser the farm, the more contribution can be made 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.

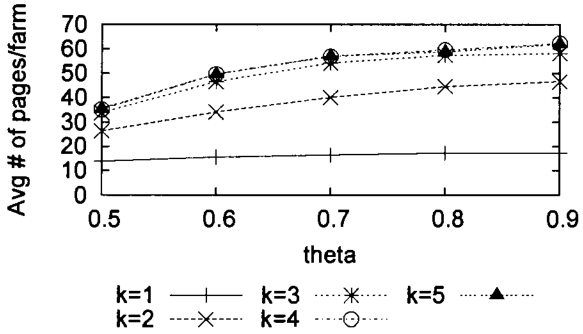


Figure 4.4: The effects of parameters θ and k on the page farm extraction (**FedEx data set**).

We checked the page farm size distribution with respect to θ and k on individual Web sites. The distribution within individual Web sites is very similar to the one on the whole sample. For example, Figure 4.4 shows the results on a sample of 4,274 Web pages from site "`http://www.fedex.com`" (called the **FedEx data set** hereafter).

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 overlap largely. In the rest of this section, by default we report results on (0.8, 3)-farms of Web pages.

We also compared the similarity of page farms extracted using the simple greedy algorithm (Section 4.1.1) and the more efficient algorithm (Section 4.1.3). The results are shown

Average similarity
0.784

Table 4.1: The average similarity of page farms extracted using the two different extraction algorithms.

in Table 4.1. For 5000 pages that we randomly selected from the data set, we extracted the $(0.8, 3)$-farms using the two algorithms. Thus for each page, we had two different page farms. Suppose the target page is p and the two page farms are $F_1(p)$ and $F_2(p)$ respectively. We calculated the similarity of the two farms based on the following formula:

$$Sim(F_1(p), F_2(p)) = \frac{|F_1(p) \cap F_2(p)|}{|F_1(p) \cup F_2(p)|}.$$

Table 4.1 shows the average similarity value of 5000 pages. Generally, the results returned by two methods are very similar.

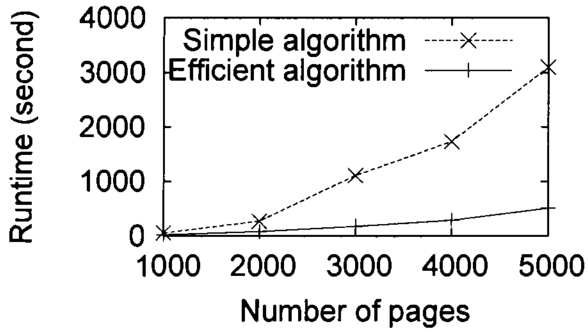


Figure 4.5: The scalability of page farm extraction.

We tested the page farm extraction efficiency using the simple greedy algorithm (Section 4.1.1) and the more efficient algorithm (Section 4.1.3). The results are shown in Figure 4.5, where the number of pages in the Web graph varies from 1,000 to 5,000, and a $(0.8, 3)$-farm for each page is extracted.

The method in Section 4.1.3 is clearly more efficient than the simple greedy method. As analyzed before, path contributions are much easier to compute. On the other hand, extracting page farms is still a time consuming task. To extract all page farms in our data set of more than 3 million pages, it took more than 20 hours using our current PC. One of our future work is to find even more efficient algorithms to extract (θ, k)-farms.

4.2.2 Page Farm Analysis on Individual Sites

To understand the general "landscapes" of page farms (that is, how page farms look like generally), we conducted the clustering analysis on the page farms extracted. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters [39]. Clustering is the process of grouping the data into clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters. Dissimilarities are assessed based on the attribute values describing the objects. Often, distance measures are used.

Since each page has a corresponding (θ, k)-farm, we treat page farms as the objects to be clustered. Our analysis was in two steps. First, we analyzed the statistics of page farms in individual sites. Then, we analyzed the statistics of page farms in the whole data set (Section 4.2.3).

Site-id	Site	# pages crawled
Site-1	http://www.fedex.com	4274
Site-2	http://www.siia.net	2722
Site-3	http://www.indiana.edu	2591
Site-4	http://www.worldbank.org	2430
Site-5	http://www.fema.gov	4838
Site-6	http://www.liverpoolfc.tv	1854
Site-7	http://www.eca.eu.int	4629
Site-8	http://www.onr.navy.mil	4586
Site-9	http://www.dpi.state.wi.us	5118
Site-10	http://www.pku.edu.cn	6972
Site-11	http://www.cnrs.fr	2503
Site-12	http://www.jpf.go.jp	5685
Site-13	http://www.usc.es	2138

Table 4.2: List of sites with different domains.

In the whole data set, there are about 50 thousand different sites and about 30 different domains[2]. In order to analyze the page farms of individual sites, we randomly selected 13 sites with different domains, as listed in Table 4.2. Those sites include some popular domains, such as .com, .net, .edu, .org and .gov, as well as some unpopular ones, such as .tv, .int and .mil. Moreover, some domains from different countries and different languages are also involved, such as .us(USA), .cn(China), .fr(France), .jp(Japan) and .es(Spain). We did clustering analysis on each individual site and examined the results in the 13 different sites.

We first generated the complete Web graph from the whole data set containing nearly 3.3 million Web pages. A normal power method [13] is used to calculate the PageRank scores. For the pages in each site, we then extracted the $(0.8, 3)$-farm using the algorithm shown in Figure 4.2.

To analyze the page farms, we extracted the following features of each farm: (1) the number of pages in the farm; (2) the total number of intra-links in the farm; and (3) the total number of inter-links in the farm. Here, intra-links are edges connecting pages in the same farm, and inter-links are edges coming from or leaving a farm. We also considered some other features, such as the average in-degrees and out-degrees, the average PageRank score, and the diameter of the farm. The clustering results on extracted page farms shown as follows are consistent. Thus, we only use the above three features as representatives to report the results here.

The above three attributes are independent with each other and each one is an important factor to reveal the characteristics of the page farms. We treated each attribute with the same importance. Thus, we normalized all attribute values into the range $[0, 1]$ in the clustering analysis. These 3 normalized attribute values form the vector space for each page farm. We applied the conventional k-means clustering [39] on extracted page farms, where Euclidian distance was adopted to measure the distance between two page farm vectors.

The k-means algorithm takes the input parameter k, and partitions a set of n objects into k clusters so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low. The algorithm proceeds as follows. First, it randomly selects k of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster.

[2] Details can be found at http://dbpubs.stanford.edu:8091/~testbed/doc2/WebBase/crawl_lists/crawled_hosts.05-2006.f

This process iterates until the criterion function converges. Typically, the squared-error criterion is used, defines as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2,$$

where E is the sum of square-error for all objects in the database, p is the point in space representing a given object, and m_i is the mean of cluster C_i (both p and m_i are multidimensional). This criterion tries to make the resulting k clusters as compact and as separate as possible.

We varied the number of clusters (that is, the value of k in k-means clustering algorithm), and compared the clusters obtained. Interestingly, if we sort all clusters according to the size (that is, the number of pages in the clusters), those small clusters are robust when the number of clusters increases. Setting the number of clusters larger tends to split the largest cluster to generate new clusters.

# clusters	C_1	C_2	C_3	C_4	C_5
2	22	4252			
3	19	103	4152		
4	19	89	543	3623	
5	19	87	230	1280	2658

Table 4.3: The number of pages in each cluster when the number of clusters varies from 2 to 5.

For example, Table 4.3 shows the number of pages in each cluster when the number of clusters varies from 2 to 5. The FedEx data set was used. By comparing the pages in the clusters, we found that the pages in C_1 are largely the same no matter how the number of clusters is set. When the number of clusters varies from 3 to 5, the clusters C_2 of different runs also largely overlap with each other.

The above observation strongly indicates that the distance from Web pages to the center of the whole data set may follow a power law distribution. To verify this, we analyzed the distances between the page farms in the site to the mean of the sample set of the site. The results are shown in Figure 4.6. The distance follows the power law distribution as expected. This clearly explains why the smaller clusters are robust and the new clusters are often splitting from the largest cluster. Moreover, the results shown here are scale-free.

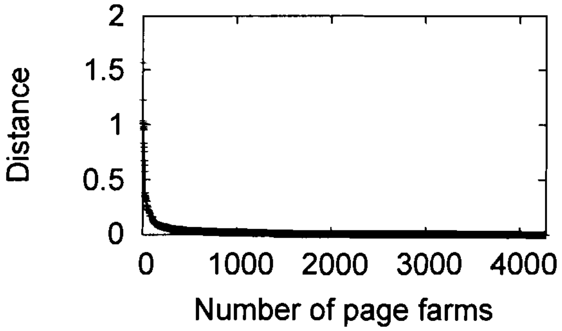


Figure 4.6: The distribution of the distance to the mean of the largest cluster.

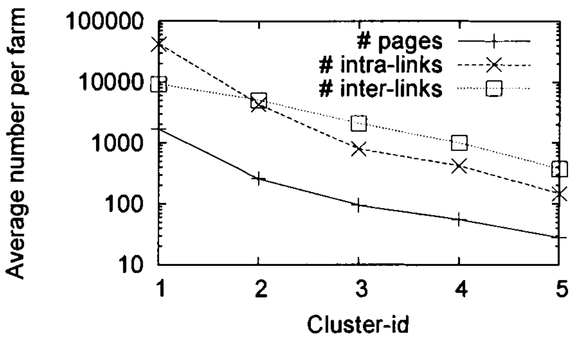


Figure 4.7: Features of page farms in clusters.

As the clusters are robust, how are the pages in different clusters different from each other? In Table 4.4, we list the top-5 URL's in each cluster that have the highest PageRank scores. Interestingly, most pages in the first cluster are the portal pages. The later clusters often have more and more specific pages of lower PageRanks. Correspondingly, In Figures 4.7, we show for each cluster the average size, the average number of intra-links, and the average number of inter-links. As can be seen, they follow the similar trend. The smaller the clusters, the larger the page farms and thus more intra- and inter-links in the farms.

Moreover, in Figure 4.8, we plot the distribution of depths of URL's in various clusters. The depth of a URL string is the number of subdirectory levels in it. For example, URL string "`http://www.fedex.com/`" is of depth 1, and URL string "`http://www.fedex.com/us/customer/`" is of depth 3. Generally, the deeper a URL, likely the more specific the

Cluster	URLs
C_1	http://www.fedex.com/
	http://www.fedex.com/us/customer/
	http://www.fedex.com/us/
	http://www.fedex.com/us/careers/
	http://www.fedex.com/us/services/
C_2	http://www.fedex.com/legal/?link=5
	http://www.fedex.com/us/search/
	http://www.fedex.com/us/privacypolicy.html?link=5
	http://www.fedex.com/us/investorrelations/?link=5
	http://www.fedex.com/us/about/?link=5
C_3	http://www.fedex.com/legal/copyright/?link=2
	http://www.fedex.com/us?link=4
	http://www.fedex.com/us/about/today/?link=4
	http://www.fedex.com/us/investorrelations/financialinfo /2005annualreport/?link=4
	http://www.fedex.com/us/dropoff/?link=4
C_4	http://www.fedex.com/ca_english/rates/?link=1
	http://www.fedex.com/legal/
	http://www.fedex.com/us/about/news/speeches?link=2
	http://www.fedex.com/us/customer/openaccount/?link=4
	http://www.fedex.com/us/careers/companies?link=4
C_5	http://www.fedex.com/?location=home&link=5
	http://www.fedex.com/ca_french/rates/?link=1
	http://www.fedex.com/ca_french/?link=1
	http://www.fedex.com/ca_english/?link=1
	http://www.fedex.com/us/careers/diversity?link=4

Table 4.4: The top-5 URLs with the highest PageRank scores in each cluster.

content. To make the comparison easy to read, we show the percentage of pages in the clusters of various depth in the figure. We can see that the pages in the first cluster (the smallest one) are not deep. The second cluster is deeper and so on. This is consistent with the observations from Table 4.4.

4.2.3 Page Farm Analysis on Multiple Sites and the Whole Data Set

The findings in Section 4.2.2 are not specific for a particular Web site. Instead, we obtained consistent observations in other Web sites, too. For example, we clustered the page farms for the 13 Web sites listed in Table 4.2 by setting the number of clusters to 5. For each site, the clusters were sorted in ascending order of the number of pages, and the ratio of the

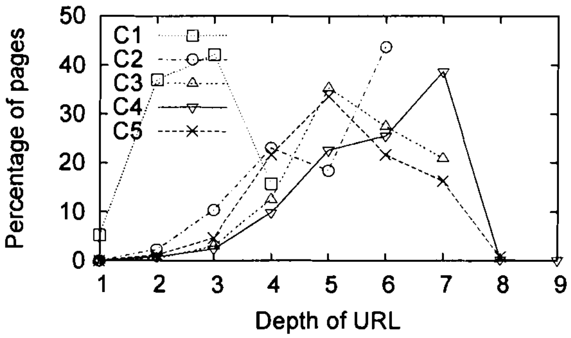


Figure 4.8: The distribution of depth of URL's in clusters.

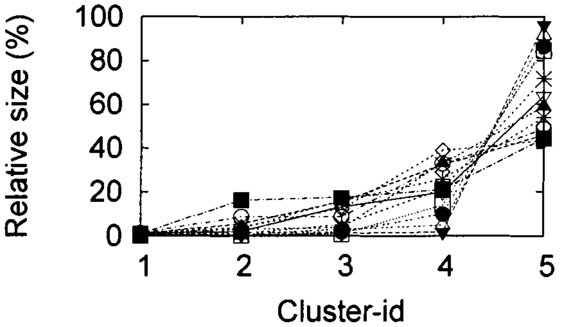


Figure 4.9: The distribution of cluster size.

number of pages in a cluster versus the total number of pages sampled from the site was used as the relative size of the cluster. Figure 4.9 shows the result. We can observe that the distributions of the relative cluster size follow the same trend in those sites.

In Section 4.2.2, we examined the page farms in individual Web sites. To test whether the properties observed were scale-free, we conducted the similar experiments on the Web page data set containing more than 3.3 million Web pages from more than 60,000 sites. For the pages in each site, we extract the (0.8, 3)-farm using the algorithm shown in Figure 4.2. The experimental results confirm that the properties are scale-free: we observed the similar phenomena on the large sample.

Figure 4.10 shows the distribution of distances of page farms to the mean of the whole data set. Clearly, it follows the power law distribution.

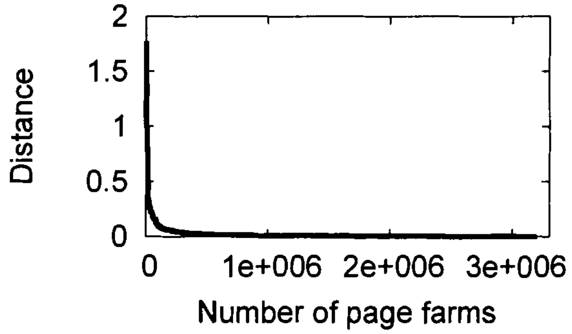


Figure 4.10: The distribution of the distance to the center of the data set.

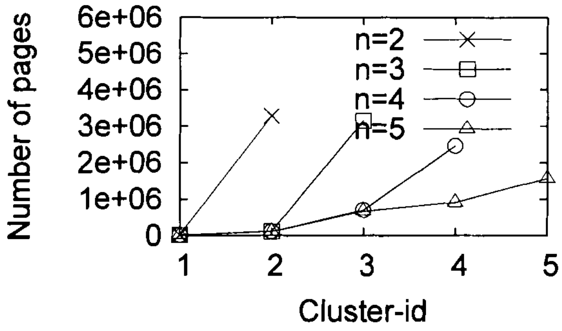


Figure 4.11: The size of clusters.

Moreover, we clustered the page farms by varying the number of clusters from 2 to 5, and sorted the clusters in size ascending order. The results are shown in Figure 4.11, where parameter n is the number of clusters. The figure clearly shows that the smaller clusters are robust and the new clusters are splitting from the largest clusters when the number of clusters is increased.

4.2.4 Summary

From the above empirical analysis of the page farms of a large sample of the Web, we can obtain the following two observations.

- *The landscapes of page farms follow a power law distribution and the distribution is scale-free.* The phenomena observed from individual large Web sites is nicely repeated on the large sample containing many Web sites across many domains.

- *Web pages can be categorized into groups according to their page farms. Some interesting features are associated with the categorization based on clustering*, such as the relative importance of the pages and the relative positions in the Web sites. The distinguishing groups are robust with respect to the clustering parameter settings.

Chapter 5

Link Spam Detection

In this chapter, we develop link spam detection methods using page farms proposed in Chapter 3, and report an empirical evaluation on a newly released spam test collection data set. The experimental results strongly show the effectiveness of our methods.

5.1 Link Spam Detection

Driven by the huge potential benefit of promoting the rankings of Web pages, many dirty tricks have been attempted to boost page rankings by making up some artificially designed link structures, which is known as *link spam* [13, 36, 43, 51]. The term "spam" here refers to any deliberate human action that is meant to trigger an unjustifiably favorable relevance or importance for some Web pages comparing to the true value of the page.

So far, the tricks of Web spam can be classified into two categories, *term spam* and *link spam* [36]. 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 structure-based ranking scores of the target page. Due to the extensive adoptions of the link structure-based ranking metrics such as PageRank [59] and HITS [48], link spam has been used deliberately by many spam pages on the Web.

Some previous studies (for example, [25, 66, 34, 9, 37, 67]) treated link 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 intension of the Web page builder, the difference on the "spamicity" (the degree of deliberation to improve the unjustifiable ranking score of a Web page) is critical to spam detection.

In this section, we propose detecting link spam using page farms. The general idea is that we can calculate a spamicity score for the page farm of a Web page to measure the likelihood of the page being link spam. In order to judge whether a page is link spam, we only need to extract the page farm of the target page. As shown in our experiments (Figure 4.3 in Chapter 4), on average the page farm of a Web page contains less than 100 pages, which can be extracted efficiently by search engines.

We explore two alternatives of defining spamicity.

5.1.1 Utility-based Spamicity

For a Web page p, let $Farm(p)$ be the page farm of p. Intuitively, if p is link spam, then $Farm(p)$ should try to achieve the PageRank score of p as high as possible. We can calculate the maximum PageRank score using the same number of pages and the same number of intra-hyperlinks as $Farm(p)$ has. The utility of the page farm of p is the ratio of the PageRank score of p against the maximum PageRank score that can be achieved. The utility can be used as a measure on the likelihood that p is link spam. The utility is in the range of $[0, 1]$. Intuitively, if the utility is closer to 1, the page is more likely to be link spam.

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

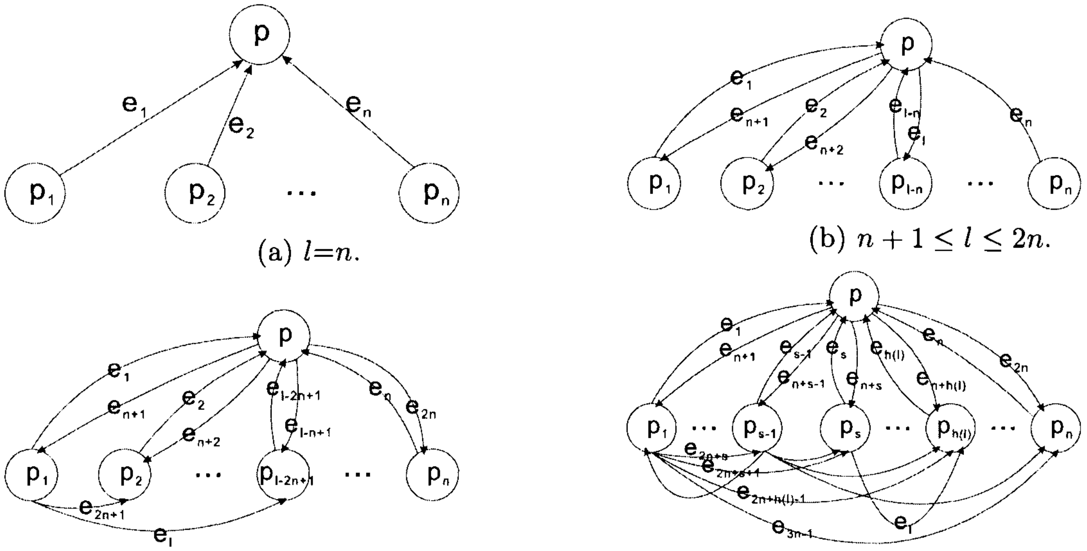
Theorem 3 (Maximum PageRank scores) *Let p be the target page, and $Farm(p)$ contains pages $p_1, \cdots, p_n$ ($p \neq p_i, i = 1, \cdots, n$) and hyperlinks $e_1, \cdots, e_l$. The following structure maximizes the PageRank score of p.*

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

where $h(i) = 1 + (i - 2n - \lceil \frac{i-2n}{n-1} \rceil (n-2) + 1)$ mod n.

Proof. Apparently, $l \geq n$ since each page in the farm must have at least one path to the target page. Generally, the optimal structures are in one of the following three cases.

First, when $l = n$, then $e_i = p_i \rightarrow p$, as shown in Figures 5.1(a).



(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, $s = \lceil \frac{l-2n}{n-1} \rceil$.

Figure 5.1: Achieving the maximum PageRank scores.

Second, when $n + 1 \leq l \leq 2n$, then, as shown in Figures 5.1(b),

$$e_i = \begin{cases} p_i \rightarrow p & (1 \leq i \leq n) \\ p \rightarrow p_{i-n} & (n+1 \leq i \leq l) \end{cases}$$

Third, when $2n + 1 \leq l \leq n(n + 1)$, in order to illustrate the way to construct the optimal structures clearly, we show a specific case in Figures 5.1(c) and the general case in Figures 5.1(d). As shown in Figure 5.1(c), when $2n + 1 \leq l \leq 3n - 1$,

$$e_i = \begin{cases} p_i \rightarrow p & (1 \leq i \leq n) \\ p \rightarrow p_{i-n} & (n+1 \leq i \leq 2n) \\ p_1 \rightarrow p_{i-2n+1} & (2n+1 \leq i \leq l) \end{cases}$$

Generally, when $2n \leq l \leq n(n + 1)$, the structure is as shown in Figure 5.1(d).

Recall the path contribution in Definition 4, given a link path P from v_1 to p: $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow p$,

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

Thus, $LCont(P, p)$ is based on two factors: the length of the link path $|P|$ and the contribution propagation $\prod_{i=1}^{n} \frac{1}{OutDeg(v_i)}$. 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.

We observe three properties of the optimal structures. First, each page p_i $(1 \leq i \leq n)$ should point to p directly. That is, for any page p_i, there is a link $p_i \rightarrow p$. In this case, the link path P from p_i to p has the smallest length $|P| = 1$ and the largest contribution propagation $\pi(P) = 1$, thus the largest path contribution.

Second, each page p_i $(1 \leq i \leq n)$ can contribute most to p if there is a circle between p_i and p. That is, for any page p_i, there are two links $p_i \rightarrow p$ and $p \rightarrow p_i$. In this way, there are infinite link paths from p_i to p, thus p obtains the maximum contribution from p_i.

Third, any link from p_i to p_j $(1 \leq i, j \leq n)$ will distract a part of contribution of p_i to p. In other words, the contribution from p_i to p will be reduced since the out-degree of p_i is increased. Thus, in order to maximize the PageRank score of p, we should avoid adding links from p_i to p_j as much as possible.

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

We first show the optimum of case (a). Since each page p_i should have a link path pointing to p, when $l = n$, each page p_i has the out-degree exactly 1 and p has the out-degree 0. In such a case, we cannot construct any circle between p_i and p. According to the first and the third observations, for each p_i, we should let p_i point to p directly, thus p can achieve the highest PageRank score. So the structure in Figure 5.1(a) is the optimal structure when $l = n$.

We next show the optimum of case (b). According to the three observations described above, we have to construct as many circles as possible between p_i and p, and construct as few links from p_i to p_j as possible. Thus, when $n+1 \leq l \leq 2n$, each page has a link pointing to p, and we can construct at most $l - n$ circles between p_i and p, and there is no link from p_i to p_j. We can conclude that the structure in Figure 5.1(b) is the optimal structure when $n + 1 \leq l \leq 2n$.

We next show the optimum of case (c). Similar to case (b), we have to construct as many circles as possible between p_i and p, and construct as few links from p_i to p_j as possible. Since $l \geq 2n + 1$, for each page p_i, we can construct a circle between p_i and p. Moreover, there are $l - 2n$ links from p_i to p_j.

We are given a target page p and its farm with n pages and l $(2n < l \leq 3n - 1)$ links.

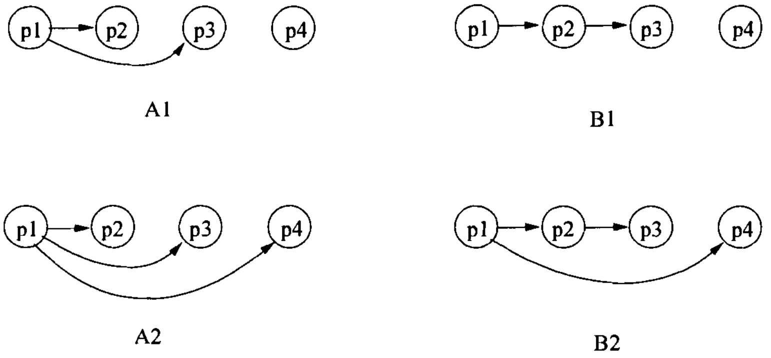


Figure 5.2: An example to illustrate the "greedy choice property" and "optimal substructure" when constructing the optimal structures.

We prove the optimum of case (c) by an induction on l, that is, the number of links in the farm.

- **Basis step**. When $l = 2n + 1$, there is one link from p_i to p_j. Obviously, any link from p_i to p_j has the same effect, thus we simply select $p_1 \rightarrow p_2$. So the structure in Figure 5.1(c) is the optimal structure when $l = 2n + 1$.

- **Inductive step**. Suppose when $l = i$ $(2n + 1 \leq i \leq 3n - 2)$, the optimal structure is shown in Figure 4(c). We want to construct the optimal structure when $l = i + 1$. As proved in [59, 51], 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. So in case (c), the sum of the PageRank scores of p and p_i $(1 \leq i \leq n)$ 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_i $(1 \leq i \leq n)$. This construction problem has the "greedy choice property" and "optimal substructure" [22], that is, the optimal farm when $l = i + 1$ can be obtained by adding one more link to the optimal farm when $l = i$, such that the increase of the sum of the PageRank scores of p_i $(1 \leq i \leq n)$ is smallest. Otherwise, we can simply use the "cut and paste" method [22] to obtain a better structure.

  Figure 5.2 gives a simple example to show the "greedy choice property" and "optimal substructure". Structures A_2 and B_2 are generated from A_1 and B_1 by adding one more link. Assume that A_1 and B_2 are the optimal structures when $l = 2n + 2$ and

$l = 2n + 3$, respectively. Suppose we use S_{A_1} to represent the sum of the PageRank scores of p_i in A_1, we have $S_{A_1} < S_{B_1}$ and $S_{A_2} > S_{B_2}$. Clearly, if we remove the link from p_1 to p_4 in B_2, the decrease of the PageRank score of p_4 is larger than that if we remove the link from p_1 to p_4 in A_2, since the out-degree of p_1 in B_2 is larger than that in A_2. Thus, if $S_{A_2} > S_{B_2}$, we have $S_{A_1} > S_{B_1}$. Contradiction.

Since the new link only can be added from p_i to p_j where $1 \leq i, j \leq n$, we want to increase the PageRank scores of p_j 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_{i-2n+1}, since the new link to p_{i-2n+1} has the length $|P| = 1$ and the smallest contribution propagation $\pi(P) = \frac{1}{i-2n+1}$. So the optimal structure when $l = i + 1$ is as shown in Figure 5.1(c).

From the basis step and the inductive step, we can conclude that the structure in Figure 5.1(c) is the optimal structure when $2n + 1 \leq l \leq 3n - 1$.

We observe that case (c) in Figure 5.1 is a special case for case (d). Thus, the optimum of case (d) can be proved in the same way.

From the above descriptions, we have Theorem 3. ■

Based on Theorem 3, 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 (Maximum PageRank scores $(n \leq l \leq 2n)$**)** *In Figure 5.1, 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. We first show the case when $l = n$. The optimal structure is shown in Figure 5.1(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 Theorem 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 4, 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 5.1(b). The way to calculate the maximum PageRank score in case (b) is as follows.

In Figure 5.1(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 Theorem 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 into the following categories: paths with length i, where $i = 1, 2, \cdots$.

We define some notations first. A link path can be denoted as $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, where $p_1, p_2, \cdots, p_n$ are the pages on this path. We use $\{p_1, p_2, \cdots, p_n\}$ to denote a set of pages. For simplicity, we use $p_1 \rightarrow \cdots \rightarrow \{p_j, \cdots, p_k\} \rightarrow \cdots \rightarrow p_n$ to denote any path $p_1 \cdots \rightarrow p_m \rightarrow \cdots \rightarrow p_n$ where $p_m \in \{p_j, \cdots, p_k\}$. We use $\mathcal{PATH}_i$ to denote the set of link paths pointing to p with length i, where $i = 1, 2, \cdots$. 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 5.1(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 $p \rightarrow p_m \rightarrow p$ to the end of $P \in \mathcal{PATH}_i$, where $p_m \in \{p_1, p_2, \cdots, p_k\}$. Thus, given a path $P \in \mathcal{PATH}_i$, we can get k paths $P \in \mathcal{PATH}_{i+2}$. According to Definition 4, we have

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

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

Thus, we have

$$\begin{aligned} PR_{max}(n,l) &= \frac{1-d}{N} + \sum_{i=1}^{\infty} LCont(\mathcal{PATH}_i) \\ &= \frac{1-d}{N} + \sum_{i=0}^{\infty} LCont(\mathcal{PATH}_{2i+2}) + + \sum_{i=0}^{\infty} LCont(\mathcal{PATH}_{2i+1}) \\ &= \frac{1-d}{N} + \sum_{i=0}^{\infty} d^{2i} LCont(\mathcal{PATH}_2) + + \sum_{i=0}^{\infty} d^{2i} LCont(\mathcal{PATH}_1) \\ &= \frac{1-d}{N} + (LCont(\mathcal{PATH}_1) + LCont(\mathcal{PATH}_2)) \times \sum_{i=0}^{\infty} d^{2i} \\ &= \frac{1-d}{N} + \frac{1}{1-d^2} \times (LCont(\mathcal{PATH}_1) + LCont(\mathcal{PATH}_2)). \qquad (5.1) \end{aligned}$$

$LCont(\mathcal{PATH}_1)$ and $LCont(\mathcal{PATH}_2)$ can be calculated as follows.

- Paths with length 1:
  - $\{p_1, \cdots, p_k\} \to p$: $\sum LCont = \frac{d(1-d)}{N} \times k$.
  - $\{p_{k+1}, \cdots, p_n\} \to p$: $\sum LCont = \frac{d(1-d)}{N} \times (n-k)$.
- Paths with length 2:
  - $p \to \{p_1, \cdots, p_k\} \to p$: $\sum LCont = d^2(1-d)\frac{1}{Nk} \times k$.

Thus, we have

$$\begin{aligned} LCont(\mathcal{PATH}_1) &= \frac{kd(1-d)}{N} + \frac{d(1-d)(n-k)}{N} \\ &= \frac{nd(1-d)}{N} \qquad (5.2) \\ LCont(\mathcal{PATH}_2) &= \frac{d^2(1-d)}{N}. \qquad (5.3) \end{aligned}$$

We apply Equation 5.2 and Equation 5.3 to Equation 5.1, then we have

$$\begin{aligned} PR_{max}(n,l) &= \frac{1-d}{N} + \frac{1}{1-d^2} \times (\frac{nd(1-d)}{N} + \frac{d^2(1-d)}{N}) \\ &= \frac{nd+1}{N(1+d)}. \end{aligned}$$

■

Corollary 4 gives the maximum PageRank scores for case (a) and case (b) in Figure 5.1 directly. However, for the other cases, there are no simple and directed ways to calculate the exact maximum PageRank scores. In our implementation, we decide to 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 5 (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

$$USpam(p) = \frac{PR(p)}{PR_{max}(|V|, |E|)}. \tag{5.4}$$

■

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 score of the target page. The spammers (that is, the builders of spam Web pages) build up the "spam farms" with the only purpose to boost the rankings of the target pages as much as possible. The optimal spam farms do not commonly happen on the Web, because they are quite different from those normal page farms.

In a typical link spam farm model, one spammer can collect some leakages from popular web sites, such as public forums and blogs. However, the spammer has little access to those source pages thus the link structures of these pages may not be consistent with the optimal structures. Moreover, since optimal spam farms are highly regular as indicated by Theorem 3, a search engine may easily detect the optimal spam farms. To disguise, a spammer may modify the optimal 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 those disguised link spam.

5.1.2 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.

Page farms are directed graphs consisting of Web pages and links. We identify three heuristics described as follows to measure the likelihood of link spam for a Web page.

Contributor Page Rank Heuristic

As indicated by the studies on authoritative pages and hub pages [48], a Web page is semantically important if it is pointed by some authoritative pages or hub pages, which often have high PageRank scores. 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 link spam.

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 6 (PageRank boosting ratio) For a target page p, let $Farm(p) = (V, E)$ be the page farm of p. The **PageRank boosting ratio** is the ratio of the PageRank of p against the average PageRank of pages in $Farm(p)$. That is,

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

■

Heuristic 1 (Contributor page rank) *The larger the PageRank boosting ratio, the more likely a page is link spam.* ■

Link Efficiency Heuristic

From Theorem 3, we can have the following result. A similar result is also observed in [35].

Corollary 5 *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 3.*

Proof. As shown in Corollary 4, 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, the 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 5.1 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 5, 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 7 (Link efficiency) For a target page p, let $Farm(p) = (V, E)$ be its page farm. The **link efficiency** of the farm is the ratio of the number of pages in $Farm(p)$ against the total number of links between the pages in V. That is,

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

■

In an average page farm that is not for spam, some random hyperlinks may exist between pages in the farm. On the other hand, in order to fully boost the target page, pages in a 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 link spam.* ■

Centralization Heuristic

In an ideal 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 pages in a 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 8 (Centralization degree) For a target page p, let $Farm(p) = (V, E)$ be its page farm. The **centralization degree** of the farm is the ratio of the indegree of p against the average indegree of the pages in $Farm(p)$. That is,

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

■

Heuristic 3 (Centralization degree) *The larger the centralization degree, the more likely a page is link spam.* ■

Characteristics-based Spamicity

Consider a virtually non-spam page p and its page farm $Farm(p)$. We have the following observations.

- The page rank boosting ratio $\beta(p)$ should approach 1 since the PageRank of p is not boosted.
- The page farm $Farm(p) = (V, E)$ should contain many pages since p is not boosted by any authoritative or hub pages. On the other hand, random hyperlink $p_i \rightarrow p_j$ happens with probability 0.5 for $p_i, p_j \in V$. Therefore, $\iota(p) = \lim_{n\rightarrow\infty} \frac{n}{\frac{n(n-1)}{2}} = 0$.
- 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.

Based on the above observations, we define the characteristics-based spamicity as follows.

Definition 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** [62]. ■

5.2 Experimental Results

To test the effectiveness of our spam detection methods using page farms, we used the recently released Webspam-UK2006 data set by the Search Engine spam project at Yahoo! Research Barcelona[1]. The data set [16] is the result of the effort of a team of volunteers. The **base data set** 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** data set consists of 8,415 different hosts chosen from the base data set. A team of volunteers were asked to classify this set of hosts as "normal", "spam" or "borderline". Moreover, the project organizers added two kinds of special votes:

[1] http://aeserver.dis.uniroma1.it/Webspam/

all the UK hosts that were mentioned in the Open Directory Project[2] 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". Intuitively, the pages in these domains are rarely spam.

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".

All the experiments were conducted on a PC computer running the Microsoft Windows XP SP2 Professional Edition operating system, with a 3.0 GHz Pentium 4 CPU, 1.0 GB main memory, and a 160 GB hard disk. The program was implemented in C/C++ using Microsoft Visual Studio. NET 2003.

5.2.1 Spamicity Distribution

We constructed the Web graph using the pages in the base data set and computed the PageRank scores of the pages. For the pages in the spam test collection, we extracted the (0.8, 3)-farms (that is, the parameters θ and k are set to 0.8 and 3 respectively).

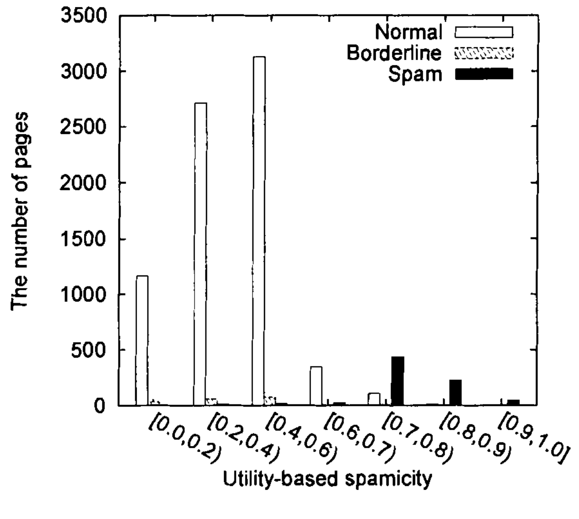


Figure 5.3: The effectiveness of spamicity in spam detection (utility-based spamicity).

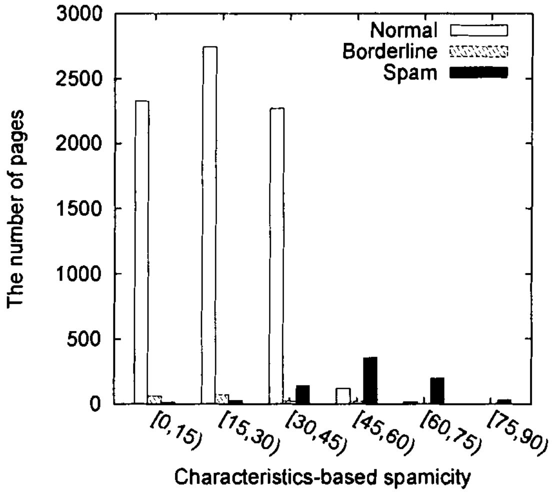
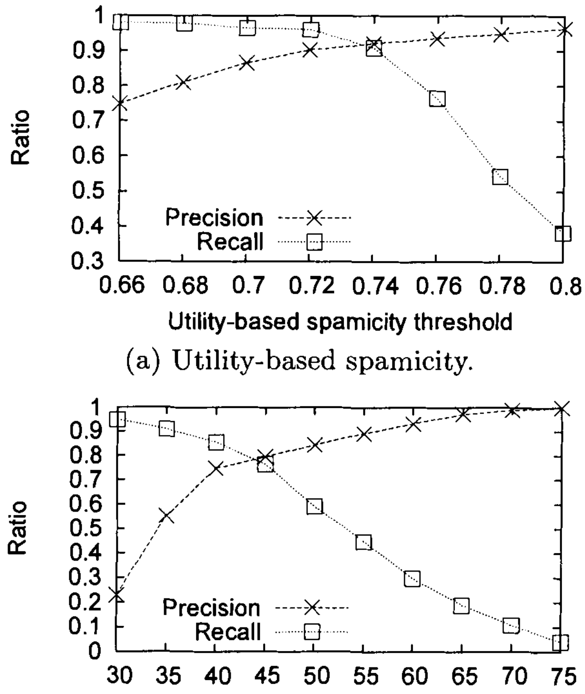


Figure 5.4: The effectiveness of spamicity in spam detection (characteristics-based spamicity).

[2] http://www.dmoz.org/

Figure 5.3 and Figure 5.4 show the distribution of normal, borderline and spam pages in the subsets of pages with various ranges of utility-based spamicity scores and characteristics-based spamicity scores, respectively. 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 data set, when the utility-based spamicity is over 0.7 and the characteristics-based spamicity is over 45, most pages are spam. This set of experiments show that the two spamicity measures can discriminate spam pages from normal ones.

5.2.2 Detecting Spam Pages



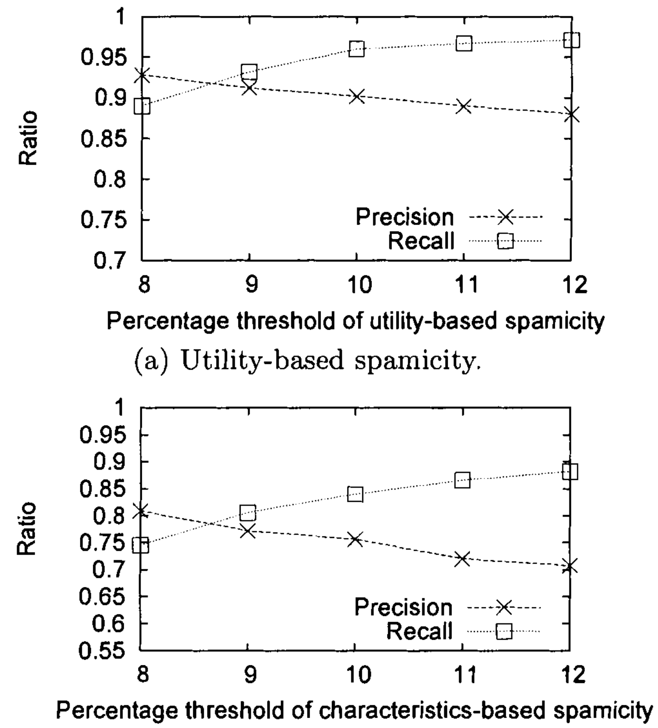
(a) Utility-based spamicity.

(b) Characteristics-based spamicity.

Figure 5.5: The precision and the recall of utility-based and characteristics-based spamicity measures (spamicity threshold).

We can simply set a spamicity threshold. The pages over the threshold are classified as spam. The pages lower than the threshold are classified as normal. Figure 5.5 shows the precision and the recall of the two spamicity measures with respect to various spamicity threshold values.

In information retrieval, precision and recall are the two basic measures used in evaluating search strategies [5]. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. These two measures are usually expressed as a percentage.



(a) Utility-based spamicity.

(b) Characteristics-based spamicity.

Figure 5.6: The precision and the recall of utility-based and characteristics-based spamicity measures (percentage threshold).

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 5.6 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 data set are spam) well. The utility-based spamicity is clearly more effective than the characteristics-based spamicity in detection quality.

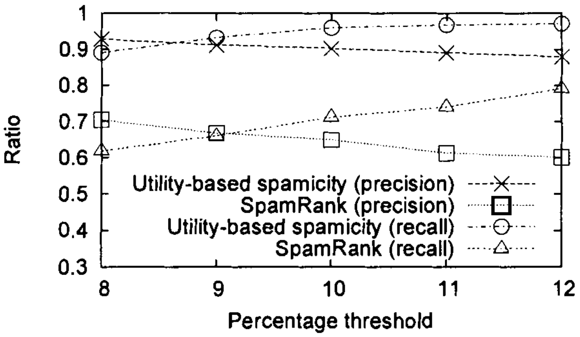
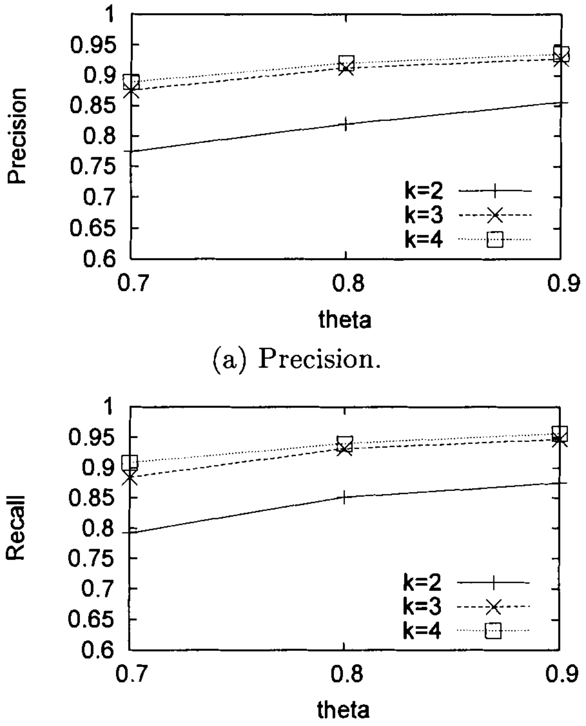


Figure 5.7: The utility-based spamicity method and SpamRank.

We also compared the utility-based spamicity method with SpamRank [9], which is the only existing method that detects link spam by assigning a spamicity-like score and does not need supervised training. SpamRank uses personalized PageRank that detects pages with an undeserved high PageRank value without the need of 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. We tried our best to implement the method as described in [9]. The results are shown in Figure 5.7. The utility-based spamicity method outperforms SpamRank in both precision and recall.

5.2.3 Effects of Page Farms in Spam Detection



(a) Precision.

(b) Recall.

Figure 5.8: The effect of page farms on spam detection accuracy.

We further examined the effects of page farms on the accuracy of the 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 5.8. We varied parameters θ and k and extracted (θ, k)-farms, then classified the top 9% of the pages of the highest utility-based spamicity as the spam pages. We measured the precision and the recall of spam detections using different (θ, k)-farms.

As discussed before, when θ and k increase, the page farms are more accurate. Figure 5.9 shows the average size of farms for those hosts in the **spam test collection** and it strongly supports our claim. Thus, the spam detection quality improves when larger and more accurate page farms are used. 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.

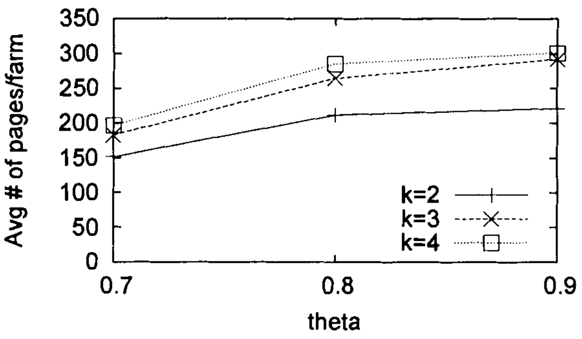


Figure 5.9: The size of farms by setting different θ and k.

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.

5.2.4 Summary

Spam detection using page farms is highly feasible and effective. 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 data set is formed, the human volunteers made judgements mostly based on the content of the pages. However, using the link analysis we can detect more than 90% of the spam. This strongly indicates that most spam pages on the real Web use both link spam and term spam.

Chapter 6

Discussions and Conclusions

As our world is now in its information era, a huge amount of data is accumulated everyday. The Web is a such kind of information collection. Though the pace has slackened since the early years (1994 – 1999), the Web continues to grow [18, 33]. Search engines are designed to help users to find useful information from the Web. They started from their IR ancestors but made a substantial technological leap. In the recent years, the information foraging on the Web is vastly easier than in the early years, but it is running up against the "syntactic search" barrier. Ranking metric is the spirit of a search engine. Since its birth with the first search engine, it has evolved from the original content-based ranking to the nowadays link structure-based ranking. However, recently search engines are facing a lot of challenging problems, such as Web spam, content evolution, and so on. Obviously, how to rank the Web pages effectively and efficiently is becoming more and more important in the Web mining and Web search areas.

In this thesis, we focus on the problem of page farm mining. We are trying to understand the essence of the rankings of Web pages. In general, we are interested in how a target page collects its ranking scores from the contributors. Understanding this question has some interesting and important applications, such as Web spam detection and Web community identification and analysis.

In this chapter, we first summarize the thesis, and then highlight the major characteristics of our page farm model. At last, we discuss some interesting and important future directions.

6.1 Summary of the Thesis

Ranking pages is an essential task in Web search. One interesting problem is, for a Web page p, what other pages are the major contributors to the ranking score of p, and how is the contribution made. In this thesis, we study the page farm mining problem and its application in link spam detection. We conclude our major contributions as follows.

- First, *we study 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 propose 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 study the computational complexity of finding page farms, and show that it is NP-hard. Then, we develop a practically feasible greedy method to extract approximate page farms.

- Second, *we empirically analyze statistics of page farms using over* 3 *million Web pages randomly sampled from the Web.* We have a few interesting and exciting findings. Most importantly, the landscapes of page farms tend to follow the power law distribution. Moreover, the landscapes of page farms strongly reflect the importance of the Web pages, and their locations in their Web sites. To the best of our knowledge, this is the first empirical study on extracting and analyzing page farms.

- Third, *we investigate the application of page farms in link spam detection.* We propose two methods. First, we measure the utility of a page farm, that is, the "perfectness" of a page farm in obtaining the maximum PageRank score, and use the utility as an index of the likeliness of link spam. Second, we use the statistics of page farms as the indicator of the likeliness of link spam. Using those measures we can detect link spam pages.

- Last, *we evaluate our link spam detection methods using a newly available real data set.* The pages are labeled by human experts. The experimental results show that our methods are effective in detecting spam pages.

6.2 Characteristics of the Page Farm Model

We have proposed a novel Web link structure model, page farm model, to understand the essence of the rankings of Web pages. We summarize the major characteristics of the page farm model here.

- *The page farm model introduces the landscapes of ecology environments for pages on the Web.* Previous work either treated each page isolated or took into account a set of pages as a whole. However, no work has been done to analyze the ecology environments of the Web, that is, the general relations of pages to their environments on the Web. The understanding of such relations has a few important applications, including Web community identification and analysis, and Web spam detection.

- *The page farm model takes into account the ranking contributions from a Web page to a target page.* Different from the simple neighborhood graphs for a given target page, in the page farm model, we rank the contributions from different Web pages and use the most important contributors to characterize the target page. The pages in the page farm have high contributions to the target page. Analyzing the page farms of Web pages can help to further understand the essence of the rankings of Web pages.

- *The page farm model adopts simple yet efficient greedy algorithm to extract approximate page farms.* The simple extraction algorithm introduced in Section 4.1.1 needs many iterations of PageRank score computation. We propose a greedy algorithm which can reduce the running time greatly, at the same time maintain good accuracy of the results.

- *The page farm model has some great potentials in Web spam detection, Web community identification, and some other important applications.* We evaluate the page farm-based link spam detection methods, and the experimental results strongly show that our methods are effective. Moreover, the experimental results also reveal that the landscapes of page farms strongly reflect the importance of the Web pages, and their locations in their Web sites.

- *The page farm model can be extended to some other ranking algorithms.* Currently, our page farm model is based on PageRank. However, we can use the same idea of page farms described in the thesis and apply it to some other ranking algorithms easily. Basically, given a target page p, the page farm of p is the contributors of the ranking score of p.

6.3 Future Work: Extensions and Applications of Page Farms

We have shown that the page farm model can be used to understand the essence of the rankings of Web pages, and to detect link spam pages on the Web effectively. Interestingly and surprisingly, the page farm model is also applicable to mining other kinds of knowledge and solving some other interesting Web mining and Web search problems. In this section, we discuss some examples.

6.3.1 Mining Core Pages and Farms from the Web

According to Theorem 2 and Corollary 3 introduced in Chapter 4, the page which has at least one out-link contributes to the PageRank scores of some other pages on the Web. One interesting question would be, given one page p, how many pages that p contributes to. Moreover, what are these pages contributing to most of the pages on the Web? Given a Web graph $G = (V, E)$ and a minimum support threshold δ, a single page which has an occurrence (that is, the number of page farms it appears in) larger than δ can be defined as a core Web page. A set of pages which are appearing in at least δ page farms at the same time can be defined as a core page set.

The core page identification problem takes into consideration only the isolated Web pages. There is not necessary to have directed links among those core pages. Another interesting question is, whether we can find those core structures among core pages on the Web. Given a Web graph $G = (V, E)$, a farm can be defined as a directed connected subgraph of G. A core farm $F = (V', E')$ is a farm such that V' is a core page set, and E' contains the directed edges induced by the pages in V'.

Mining core pages and farms has some interesting and useful applications. For example, we may understand the Web link structures further. The core farms consist of those "strongly" connected communities on the Web. Moreover, analyzing those core farms over time may help to understand the Web evolution.

6.3.2 Web Page Classification Using Page Farms

Automatic classification and categorization of Web pages is an important issue in the design of search engines. It is critical in Web information extraction. Simple text-based approaches are typically used nowadays, but most of the information provided by the page layout and link structure is discarded. Only some visual features, such as the font face and size, are effectively used to weigh the importance of the words in the page. Thus, text-based approaches sometimes cannot obtain good results.

Recently, some researchers are tying to classify Web pages based on their link structures [19, 61, 38]. As a new research topic, the effectiveness and the efficiency of link structure-based Web page classification methods still need substantial development.

Our page farm model has some potentials to classify Web pages as well. If two page farms F_p and F_q are very similar to each other, are page p and q similar to each other, too? Our empirical analysis in Chapter 4 shows some probability of positive answer to this question. Thus, we may use page farms to classify Web pages.

6.3.3 Detecting Term Spam Pages

We already show the effectiveness of our page farm-based link spam detection methods in detecting link spam. As Gyöngyi et al. concluded in [36], generally, the current Web spam technologies can be classified into two categories, link spam and term spam. Term spam is the trick which is the practice of "engineering" the content of Web pages so that they appear relevant to popular searches.

In evaluating textual relevance, search engines consider the fields, which are the locations on a Web page where the query terms occur. Each type of location is called a field. The common text fields for a page p on the Web are the document body, the title, the meta tags in the HTML header, and page p's URL. In addition, the anchor texts associated with URLs that point to p are also considered belonging to page p (anchor text field), since they often describe very well the content of p. The terms in the text fields of p are used to determine the relevance of p with respect to a specific query which is a group of query terms, often with different weights given to different fields.

Clearly, our page farm-based spam detection methods can be extended to detect term spam as well. For example, we can define the utility-based term spamicity for each page so as to find those pages which have extremely high term spamicity scores. Such pages can be classified into term spam pages. Moreover, we can combine the link-based spamicity with the term-based spamicity so as to obtain a global spamicity score for a Web page. This would be an interesting and effective way to detect spam pages on the Web thoroughly.

6.3.4 Other Future Directions

More Efficient Extraction Algorithms

In order to extract the page farms, we propose a simple yet efficient greedy algorithm. Comparing to the simple extraction algorithm introduced in Section 4.1.1, the greedy algorithm in Section 4.1.3 outperforms in its running time but maintains good accuracy of the results. However, as we can see the comparison in Figure 4.5, the page farm extraction using current greedy algorithm is still a time-consuming task. The running time is generally linear to the number of pages for which we want to extract the page farms. One of our future work is to design even more efficient algorithms for the page farm extraction.

Web Evolution Analysis based on Page Farms

The Web is a constantly evolving dynamic information collection. One important and interesting research topic is to examine the change and evolution of the Web. The questions we are interested in include "how much new content is introduced to the Web every day?", "how often do Web pages change?", "how can we model the changes of Web pages?", etc.

Some previous work has been done to evaluate the evolutions of the Web. For example, Ntoulas et al. [57] found that existing pages are being removed from the Web and replaced by new ones at a very rapid rate. However, new pages tend to "borrow" their content heavily from existing pages. The minority of pages that do persist over extended periods of time typically exhibit very little substantive change, although many undergo superficial changes.

It would be interesting to investigate the evolutions of the Web based on page farm analysis. Since page farms reveal the ecology environments of Web pages, the changing of environments would reveals some information about the changing of pages.

New Web Page Ranking Metrics

In recent years, search engines are facing many challenging problems, such as Web spam, content evaluation, and so on. How to rank the pages efficiently and effectively is becoming more and more important. Even more, some previous work reveals that the current ranking metrics adopted by search engines have their bias. For example, in [20], Cho et al. investigated the problem of page quality. In [20], they discussed how to quantify the subjective notion of page quality, how well the existing search engines measure the quality, and how they might measure the quality of a page more directly. As a conclusion, they introduced the "rich gets richer" phenomena on the Web due to the search engine bias. It is on demand to design new ranking metrics for the new generation of Web search engines.

Page farms provide more comprehensive information than the target pages themselves. We may design the ranking metrics by considering not only the pages themselves, but also those pages in their page farms. Such kind of ranking metrics would be robust.

The Web has become a vast storehouse of information and knowledge, which is built in a decentralized yet collaborative manner. Web mining – "the automatic discovery of interesting and valuable information from the Web" [18] – has therefore become an important theme in data mining area. It is the data miner's responsibility to distinguish the gold from the dust.

Bibliography

[1] James Abello, Adam L. Buchsbaum, and Jeffery R. Westbrook. A functional approach to external graph algorithms. In Gianfranco Bilardi, Giuseppe F. Italiano, Andrea Pietracaprina, and Geppino Pucci, editors, *Proceedings of the 6th Annual European Symposium on Algorithms (ESA '98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 332–343, London, UK, August 1998. Springer-Verlag.

[2] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. The diameter of the world wide web. *Nature*, 401:130–131, 1999.

[3] David Aldous. *Random Walks on Finite Groups and Rapidly Mixing Markov Chains.* Springer-Verlag, Berlin, 1983.

[4] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, pages 44–54, New York, NY, USA, 2006. ACM Press.

[5] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press/Addison-Wesley, 1999.

[6] Pierre Baldi, Paolo Frasconi, and Padhraic Smyth. *Modeling the Internet and the Web: Probabilistic Methods and Algorithms.* Wiley, 2003.

[7] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(15):509–512, 1999.

[8] J. A. Barnes. Class and committees in a norwegian island parish. *Human Relations*, 7:39–58, 1954.

[9] Andras A. Benczur, Karoly Csalogany, Tamas Sarlos, and Mate Uher. Spamrank: Fully automatic link spam detection. In *Proceedings of the 1st International Workshop on Adversarial InformationRetrieval on the Web (AIRWeb'05)*, 2005.

[10] Michael K. Bergman. The deep web: Surfacing hidden value. http://www.brightplanet.com/resources/details/deepweb.html.

[11] Krishna Bharat and Monika R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st ACM International Conference on Researchand Development in Information Retrieval (SIGIR'98)*, pages 104–111, Melbourne, AU, 1998.

[12] Zhiqiang Bi, Christos Faloutsos, and Flip Korn. The "dgx" gistribution for mining massive, skewed data. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 17–26, New York, NY, USA, 2001. ACM Press.

[13] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Transactions on Internet Technology*, 5(1):92–128, 2005.

[14] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th International Conference on World Wide Web (WWW'00)*, pages 309–320. North-Holland Publishing Co., 2000.

[15] R. S. Burt and M. Minor. *Applied Network Analysis: A Methodological Introduction.* Sage, Beverly Hills, 1983.

[16] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, 2006.

[17] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM'04)*, Philadelphia, PA, 2004. SIAM.

[18] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from HyperText Data.* Science and Technology Books, 2002.

[19] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD'98)*, pages 307–318, New York, NY, USA, 1998. ACM Press.

[20] Junghoo Cho, Sourashis Roy, and Robert E. Adams. Page quality: in search of an unbiased web ranking. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD'05)*, pages 551–562, New York, NY, USA, 2005. ACM Press.

[21] J. Coleman. *Foundations of Social Theory.* Harvard University Press, Harvard, 1990.

[22] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* McGraw-Hill Higher Education, 2001.

[23] Reinhard Diestel. *Graph Theory (3rd Edition)*, volume 173. Springer-Verlag, Heidelberg, 2005.

[24] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power law relationships of the internet topology. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'99)*, pages 251–262, New York, NY, USA, 1999. ACM Press.

[25] Dennis Fetterly, Mark Manasse, and Marc Najork. 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)*, pages 1–6, New York, NY, USA, 2004. ACM Press.

[26] Dennis Fetterly, Mark Manasse, and Marc Najork. 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)*, pages 170–177, New York, NY, USA, 2005. ACM Press.

[27] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1:385–408, 2004.

[28] Gary Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, pages 150–160, Boston, MA, August 20–23 2000. ACM.

[29] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[30] L. C. Freeman, D. R. White, and A. K. Romney. *Research Methods in Social Network Analysis.* George Mason University Press, Fairfax, VA, 1989.

[31] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *UK Conference on Hypertext*, pages 225–234, 1998.

[32] Michelle Girvan and M. Newman. Community structure in social and biological networks. *PROC.NATL.ACAD.SCI.USA*, 99:7821, 2002.

[33] Antonio Gulli and Alessio Signorini. The indexable web is more than 11.5 billion pages. http://www.cs.uiowa.edu/ asignori/web-size.

[34] Zoltán Gyöngyi, Pavel Berkhin, Hector Garcia-Molina, and Jan Pedersen. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Databases (VLDB'06)*, pages 439–450. ACM, 2006.

[35] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Databases (VLDB'05)*, pages 517–528. ACM, 2005.

[36] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05)*, 2005.

[37] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the 30th International Conference on Very Large Databases (VLDB'04)*, pages 576–587. Morgan Kaufmann, 2004.

[38] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Web content categorization using link information. Technical report, Stanford University, 2006.

[39] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers, 2003.

[40] Robert A. Hanneman and Mark Riddle. *Introduction to Social Network Methods.* University of California, Riverside, 2005.

[41] T. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11st International World Wide Web Conference (WWW'02)*, pages 784–796, Honolulu, Hawaii, 2002. ACM.

[42] T. Haveliwala, A. Gionis, D. Klein, and P. Indyk. Evaluating strategies for similarity search of the web. In *Proceedings of the 11st International World Wide Web Conference (WWW'02)*, pages 432–442, Honolulu, Hawaii, 2002. ACM.

[43] M. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1573–1579, 2003.

[44] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Natural communities in large linked networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pages 541–546, New York, NY, USA, 2003. ACM Press.

[45] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12nd International World Wide Web Conference (WWW'03)*, pages 271–279, Budapest, Hungary, 2003. ACM.

[46] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 1972.

[47] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pages 137–146, New York, NY, USA, 2003. ACM Press.

[48] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA'98)*, pages 668–677. ACM, 1998.

[49] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17, 1999.

[50] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. In *Proceeding of the 8th International Conference on World Wide Web (WWW'99)*, pages 1481–1493, New York, NY, USA, 1999. Elsevier North-Holland, Inc.

[51] A. Langville and C. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.

[52] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceeding of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD'05)*, pages 177–187, New York, NY, USA, 2005. ACM Press.

[53] P. R. Monge and N. S. Contractor. *Emergence of Communication Networks.* Sage, Thousand Oaks, CA, 2006. New Handbook of Organizational Communication.

[54] Rajeev Motwani and Ying Xu. Evolution of page popularity under random web graph models. In *Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'06)*, pages 134–142, New York, NY, USA, 2006. ACM Press.

[55] Isheeta Nargis, David A. Pike, and Neil McKay. Neighborhoods in the web graph. In *Proceedings of the 4th Workshop on Algorithms and Models for the Web Graph (WAW'06)*, 2006.

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

[57] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*, pages 1–12, New York, NY, USA, 2004. ACM Press.

[58] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International World Wide Web Conference (WWW'06)*, pages 83–92, New York, NY, USA, 2006. ACM Press.

[59] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[60] John Scott. *Social Network Analysis Handbook.* Sage Publications Inc., 2000.

[61] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. A comparison of implicit and explicit links for web page classification. In *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*, pages 643–650, New York, NY, USA, 2006. ACM Press.

[62] A. C. Thompson. *Minkowski Geometry.* Cambridge University Press, New York, 1996.

[63] Stanley Wasserman and Katherine Faust. *Social Network Analysis.* Cambridge University Press, 1994.

[64] B. Wellman and S. D. Berkowitz. *Social Structures: A Network Approach.* Cambridge University Press, Cambridge, 1988.

[65] Barry Wellman. For a social network analysis of computer networks: a sociological perspective on collaborative work and virtual community. In *Proceedings of the 1996 ACM SIGCPR/SIGMIS Conference on Computer Personnel Research (SIGCPR'96)*, pages 1–11, New York, NY, USA, 1996. ACM Press.

[66] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference (WWW'05)*, pages 820–829, New York, NY, USA, 2005. ACM Press.

[67] Baoning Wu, Vinay Goel, and Brian D. Davison. Topical trustrank: Using topicality to combat web spam. In *Proceedings of the 15th International World Wide Web Conference (WWW'06)*, pages 63–72, New York, NY, USA, 2006. ACM Press.

[68] Ricardo Baeza Yates, Paolo Boldi, and Carlos Castillo. 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)*, pages 308–315, New York, NY, USA, 2006. ACM Press.

[69] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy. Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web Graph (WAW'04)*, volume 3243 of *Lecture Notes in Computer Science*, pages 92–104. Springer, October 2004.