Identifying Web Spam with the Wisdom of the Crowds

YIQUN LIU, FEI CHEN, WEIZE KONG, HUIJIA YU, MIN ZHANG, SHAOPING MA, LIYUN RU

State Key Lab of Intelligent Technology & Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China P.R.

Combating Web spam has become one of the top challenges for Web search engines. State-of-the-art spam detection techniques are usually designed for specific, known types of Web spam and are incapable and inefficient for newly-appeared spam types. With user behavior analyses from Web access logs, a spam page detection algorithm is proposed based on a learning scheme. The main contributions are the following: (1) User visiting patterns of spam pages are studied, and a number of user behavior features are proposed to separate Web spam pages from ordinary pages. (2) A novel spam detection framework is proposed that can detect various kinds of Web spam including newly-appeared ones with the help of the analysis of user behavior. Experiments on large scale practical Web access log data show the effectiveness of the proposed features and the detection framework..

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Search process; H.4 [**INFORMATION SYSTEMS APPLICATIONS**]; H.5.4 [**Hypertext/Hypermedia**]: User issues

General Terms: Measurement, Experimentation, Human Factors

Additional Key Words and Phrases: Spam detection, Web search engine, User behavior analysis

---

1. INTRODUCTION

With the explosive growth of information on the Web, search engines have become more and more important in people's daily lives. According to a search behavior survey report in China, 69.4% of internet users utilize search engines, and 84.5% regard using search engines as a major way to access Web information [CNNIC 2009]. Although search engines usually return thousands of results for a query, most search engine users only view the first few pages in result lists [Silverstein et al. 1999]. As a consequence, ranking position of the results has become a major concern of internet service providers.

This research was supported by Natural Science Foundation of China (60736044, 60903107) and Research Fund for the Doctoral Program of Higher Education of China (20090002120005).

Authors' addresses: FIT Building 1-506, Tsinghua University, Beijing, China, 100084; Email: yiqunliu@tsinghua.edu.cn.

To obtain "an unjustifiably favorable relevance or importance score for some Web page, considering the page's true value" [Gyongyi and Garcia-Molina 2005], various type of Web spam techniques have been designed to mislead search engines. In 2006, it was estimated that approximately one seventh of English Web pages were spam, and these spam pages became obstacles in users' information acquisition process [Wang et al. 2007]. Therefore, spam detection is regarded as a major challenge for Web search service providers [Henzinger et al. 2003].

Most anti-spam techniques make use of Web page features, either content-based or hyper-link structure based, to construct Web spam classifiers. In this kind of spam detection framework, when a certain type of Web spam appears in search engine results, anti-spam engineers examine its characteristics and design specific strategies to identify it. However, once one type of spam is detected and banned, spammers develop new spam techniques to cheat search engines. Since search engines' wide adoption in the late 1990s, Web spam has evolved from term spamming and link spamming to the currently used hiding and JavaScript spamming techniques. Although machine learning based methods have shown their superiority by being easily adapted to newly-developed spam, these approaches still require researchers to provide a specific spam page's features and build up suitable training sets.

This kind of anti-spam framework has caused many problems in the development of Web search engines. Anti-spam has become an ever-lasting process. It is quite difficult for anti-spam techniques to be designed and implemented in time because when the engineers become aware of a certain spam type, it has already succeeded in attracting many users' attention, and spammers have turned to new types of spam techniques.

In contrast to the prevailing approaches, we propose a different type of anti-spam framework: the Web Spam Detection framework based on the wisdom of the crowds. Recently, the wisdom of the crowds has gained much attention in Web search research (see, e.g., Fuxman et al. [2008], Bilenko and White [2008]). In these works, wisdom of the crowds is usually considered as a kind of implicit feedback information for page relevance and importance. Different from the previous works, we adopt the wisdom of the crowds to identify spam pages by the analysis of users' Web access logs. Web spam attempts to deceive a search engine's ranking algorithm instead of meeting Web user's information needs as ordinary pages. Therefore, the user-visiting patterns of Web spam pages differ from those of ordinary Web pages. By collecting and analyzing large-scale user-access data of Web pages, we can find several user behavior features of spam pages.

These features are used to develop an anti-spam algorithm to identify Web spam in a timely, effective, and type-independent manner.

The contributions of this paper are the following:

1. We propose a Web spam detection framework in which spam sites are identified based on their deceitful motivation instead of their content/hyper-link appearance.
2. We propose six user-behavior features extracted by the analysis of users' Web access logs. These features can distinguish spam Web sites from ordinary sites quickly and effectively.
3. We designed a learning-based approach to combine the proposed user-behavior features to compute the likelihood that the Web sites are spam. Differently from traditional algorithms, this naïve Bayes based approach employed positive examples as well as unlabeled data to finish the spam detection task.

The remainder of the paper is organized as follows. Section 2 gives a brief review of related work in Web spam detection. Section 3 analyzes the differences in user-visiting patterns between Web spam and ordinary pages and proposes corresponding features. The spam detection framework based on behavior analysis and a learning scheme is proposed in Section 4, and experimental results are presented in Section 5 using a performance evaluation on large scale practical Web access logs. The paper ends with the conclusion and a discussion of future work.

2. RELATED WORK

2.1 Web Spamming Techniques

According to Gyongyi's Web spamming taxonomy proposed in [Gyongyi and Garcia-Molina 2005], spamming techniques are grouped into two categories: term spamming and link spamming.

Term spamming refers to techniques that tailor the contents of special HTML text fields to make spam pages relevant for some queries [Gyongyi and Garcia-Molina 2005]. HTML fields that are often adopted by spamming techniques include page titles, keywords in the Meta field, URLs and hyper-link anchors. Hot search keywords are listed (sometimes repeatedly) in these fields to obtain high rankings by cheating search engines' content relevance calculation algorithms.

Link spammers create hyper-link structures to optimize their scores in hyper-link structure analysis algorithms. Link analysis algorithms, such as PageRank [Brin and Page 2008] and HITS [Kleinberg 1999], are usually adopted to evaluate the importance of Web

pages. Link farms, honey pots and spam link exchange are all means of manipulating the link graph to confuse these algorithms.

After Gyongyi's spam taxonomy [Gyongyi and Garcia-Molina 2005] was proposed in 2005, many more spam types appeared on the Web, and it is difficult to group some of these types into the proposed categories. Spam pages' content crawled by Web search spiders may differ from what users see because of cloaking techniques [Wu and Davison 2005]. Browsers may be redirected to visit third-party spam domains when users want to browse "normal" pages [Wang et al. 2007]. JavaScript, Cascading Style Sheets (CSS) or even Flash movies are currently being adopted by spammers (See Figure 1).

Fig. 1. A Web spam page that uses JavaScript to hide ads. The cell phone ring tone download ads are hidden in the JavaScript http://www.xinw.cn/10086/go1.js.
Left: HTML text of the page; Right: appearance of the page.

2.2 Web spam detection algorithms

Once a new type of Web spam appears on the Web, an anti-spam technique will be developed to identify it. New Web spam techniques will then be implemented to confuse that technique and so on. To combat Web spam and improve the search user experience, search engines and Web search researchers have developed many methods to detect Web spam pages. Recently, Castillo and Davison [2010] gave a comprehensive review on spam fighting and they grouped existing techniques into content-based, link-based and usage-data-based ones.

2.2.1 Content-based and link-based spam detection algorithms

Sometimes spamming activities can be detected by the analysis of content-based statistical features of page contents, such as those described in the works of Fetterly et al. [2004] and Ntoulas et al. [2006]. Recently, Cormack et al. [2011] built a classifier from honeypot queries with N-gram content-based features. They found that the classifier

worked well on the terabyte scale Web corpus ClueWeb 1 and improved retrieval performance for most TREC[2] Web track results.

Meanwhile, most Web spam identification efforts have focused on hyper-link structure analysis. The works Davison [2000] and Amitay et al. [2003] were among the earliest studies of Web link spam. Gyongyi et al. [2004] proposed the TrustRank algorithm to separate reputable pages from spam. His work was followed by large efforts in spam page link analysis, such as Anti-Trust Rank [Krishnan and Raj, 2006] and Truncated PageRank [Becchetti et al. 2006]. Learning-based methods were also adopted to combine hyperlink features to obtain better detection performance [Geng et al. 2007]. Wu and Davison [Wu and Davison 2005] proposed an anti-cloaking method by crawling and comparing different copies of a Web page. Wang et al. [2007] proposed identifying redirection spam pages by connecting spammers and advertisers through redirection analysis. Svore et al. [2007] adopted query-dependent features to improve spam detection performance.

These anti-spam techniques can detect specific types of Web spam, and most can achieve good identification performance. However, because there are always new types of spamming techniques, Web spam can still be found in a search engine's result lists, sometimes at high ranking positions. There are two major problems with these spam detection methods:

1. The "multi-type problem": most state-of-the-art anti-spam techniques are designed to deal with a single type of Web spam; this complicates a search engine's anti-spam process because it has to identify all current types of spam.
2. The "timeliness problem": although anti-spam techniques adopted by search engines can identify many types of Web spam, how a newly-appeared kind of Web spam is identified at an early stage before it disrupts search users still remains a problem.

2.2.2 Spam detection with usage data analysis

In order to solve these two problems with existing content-based or link-based algorithms, some researchers tried to improve search engines' spam detection performance by user behavior analysis. These works relied on data from search logs, browsing logs or ad-click logs to identify spammers or spamming activities. Many of these works focused on the identification of click fraud [Jansen 2007] or automatic search traffic [Buehrer 2008] by separating abnormal clicks from ordinary ones. They concerned

[1] http://boston.lti.cs.cmu.edu/Data/clueweb09/

more about removing spamming activities in usage data than removing Web spam pages with the help of usage data. In the research field of spam detection with usage data analysis, Bacarella et al. [2004] constructed a traffic graph with browsing behavior data and found that sites with very high relative traffic were usually Web spam. Ntoulas et al. [2006] and Castillo et al. [2008] used search query log analysis to locate honeypot terms that are usually employed by spammers. Chellapilla and Chickering [2006] further found that both popular queries and highly monetizable queries could be chosen as honeypot terms. The major difference between our spam detection framework and these existing techniques is that we focused on both search behavior data and browsing behavior data. This gives us a clear picture on how users are led to spam pages and how users interact with these pages. By this means, user visiting patterns of both spam pages and ordinary pages are compared and corresponding spam detection method is designed with these patterns.

In our previous work [Liu et al. 2008a][Liu et al. 2008b], we proposed three user behavior features and constructed a Bayes classifier to separate spam from ordinary pages. This paper proposes three new features that are derived from users' behavior information. The performance of the proposed algorithm is also compared with some widely-adopted learning algorithms. In addition, we used a different data set from the one used previously in [Liu et al. 2008a][Liu et al. 2008b]. Our goal was to prove the robustness and effectiveness of this spam detection method.

3. USER BEHAVIOR DATA SET

3.1 Web Access Logs

With the development of search engines, Web browser toolbars have become more and more popular. Many search engines develop toolbar software to attract more user visits (e.g., Google and Yahoo). Web users usually adopt toolbars to obtain instant access to search engine services and to obtain browser enhancements such as pop-up window blocking and download acceleration. To provide value-added services to users, most toolbar services also collect anonymous click-through information from users' browsing behavior with the permission of user agreement licenses. Previous works [Bilenko and White 2008] used this kind of click-through information to improve ranking performance. In this paper, we adopted Web access logs collected by search toolbars because this type

[2] http://trec.nist.gov/

of data source collects user behavior information at a low cost without interrupting the users' browsing behavior. Information recorded in Web access logs is shown in Table I.

Table I. Information recorded in Web access logs

Name	Description
User ID	A randomly assigned ID for each user[3]
Source URL	URL of the page that the user is visiting
Destination URL	URL of the page that the user navigates to
Time stamp	Time when the Web browsing behavior occurs

Example 1. Web access log sample collected on Dec. 15th, 2008

Time stamp	User ID	Source URL	Destination URL
01:07:09	3ffd50dc34fcd7409100101c63e9245b	http://v.youku.com/v_playlist/f1707968o1p7.html	http://www.youku.com/playlist_show/id_1707968.html
01:07:09	f0ac3a4a87d1a24b9c1aa328120366b0	http://user.qzone.qq.com/234866837	http://cnc.imgcache.qq.com/qzone/blog/tmygb_static.htm
01:07:09	3fb5ae2833252541b9ccd9820bad30f6	http://www.qzone8.net/hack/45665.html	http://www.qzone8.net/hack/

From Table I and Example 1, we can see that no privacy information was included in the log data. The information shown can be easily recorded using browser toolbars by commercial search engine systems. Therefore, it is practical and feasible to obtain these types of information and to apply them in Web spam detection. With the help of a widely used commercial Chinese search engine, Web access logs were collected from Nov. 12th, 2008 to Dec. 15th, 2008. Altogether, 3.49 billion user clicks on 970 million Web pages (4.25 million sites) and 28.1 million user sessions were recorded in these logs.

3.2 Data cleansing for Web Access Logs

After collecting the Web access logs, a data cleansing process is needed to reduce possible noise. We performed the following three steps to retain the meaningful user behavior data.

3.2.1 Redirection detection

In the Web access logs, some clicks are not actually performed by users. Instead, they are caused by automatic redirection links. These records should be reduced because they are not "user behavior" data. We identified redirection links by frequently-appearing link patterns and verified these patterns using a Web crawler. In this process, about 70 million redirection records were removed from the Web access logs. Although some previous works [Buehrer et al. 2008] pointed out that redirection behavior can be used as a sign for spam behavior, we found that most redirections come from ordinary Web pages. For example, the site http://www.g.cn/ is redirected to http://www.google.cn/ because although the first URL is short and easy to remember for most Chinese users, the latter site actually provides search engine services.

3.2.2 Click fraud detection

When we analyzed Web access logs, we assumed that each click meant that the user wanted to visit the destination page to obtain information or services. However, users may follow links without having an actual interest in the target page. They may click ads for the purpose of generating a charge in pay per click advertising, an activity called click fraud. In this process, we detected about 41 million click fraud records using techniques of the commercial search engine that helped us collect the Web access logs.

3.2.3 UV and UV from unique IP

If one Web site receives few user visits (UV for short) or if most of the UVs are from a single IP, we can see that the user behavior data for this site comes from only a few users. This kind of behavior data may be biased and unreliable. Therefore, we should reduce behavior data for these Web sites. We looked into the access logs and calculated the UV data for each Web site recorded. In this way, user behavior data for Web sites with a UV less than 10 (including 1.01 billion user clicks) were removed.

Approximately 68% of the data were retained after 1.12 billion user clicks were removed in these three steps. We believe that it is necessary to perform the data cleansing process because effective user behavior features cannot be extracted from a dataset filled with noisy and unreliable data.

3.3 Construction of Spam Training and Test Sets

During the time period in which the access log was collected, the spam training set was also constructed by three professional assessors. At first, these assessors examined the

[3] The user ID is assigned by the browser toolbar software. It remained the same as long as the browser's cookie information is not emptied. When that happens, a new user ID is re-assigned.

search result lists of 500 frequently-proposed queries. These queries were random sampled from the hottest ones that were submitted to the search engine that collected access log. Navigational type queries were removed because few spam pages appear in their corresponding search result lists.

In the training set annotation process, one page P is annotated as a spam page if it meets any of the following criteria:

- For a certain query Q, P is in the first result page of Q, while the page quality of P or the relevance of P with Q is much lower than the result should be.
- P contains deceitful or illegal content, such as gambling or pornography content.
- P's content is obviously not the same as it appears in the search result list.
- P contains malicious software that may infect users' computers.

Two assessors first examined the search result lists and annotated a number of spam pages. When their opinions about certain pages had conflicts, the third assessor decided whose annotation result was accepted. After that, because we needed a site-level training set, the annotated spam pages were examined again to see whether their corresponding sites could be regarded as a spam site (most pages in a spam site should be spam pages). Again, the same three assessors finished the site level annotation task. We also reduced the sites that didn't appear in the cleansed Web access logs. Finally, we collected 802 sites for the Web spam training set.

For the spam detection test set, we randomly sampled 1,997 Web sites from the cleansed access log (about 1/2000 of all Web sites covered in the corpus) and had the same three assessors annotate these sites. The annotation process of spam training set was directed by hot non-navigational queries (including many honeypot queries) while the test set was constructed by random sampling. The reason is that training process could be based on spam (positive) examples and unlabeled data while test process should involve both spam (positive) and non-spam (negative) examples to estimate performance of the proposed algorithm. If we also used search lists of some honeypot queries to construct the test set, the sampling process would be biased both in contents and in page qualities (most top-ranked results are high quality ones). For a candidate Web site S in the test set, the annotation process is performed as follows:

Firstly, we extracted user visiting information of the pages in S from the Web access logs and listed all pages together with its user visiting frequencies. Secondly, the annotators examined the first few most frequently visited pages and checked whether they were spam pages. one page P is annotated as a spam page if it meets any of the following criteria:

- P doesn't contain the information that it declares to have in its page title or title of its main body.
- P's content cannot be read due to language mistakes or too many advertisements.
- P contains deceitful or illegal content, such as gambling or pornography content.
- P contains malicious software that may infect users' computers.

Two assessors first examined the search result lists and annotated a number of spam pages. When their opinions about certain pages had conflicts, the third assessor decided whose annotation result was accepted. If three or more most frequently visited pages of S is annotated as "spam", S is annotated as a spam site. The annotation result was that 491 sites were spam, 1248 were non-spam, and assessors "could not tell" whether 258 sites were spam or not because these sites could not be connected at the time of annotation.

With these access logs and the spam training/test set, we were able to investigate the different behavior patterns between ordinary and spam pages to better understand the perceptual and cognitive factors underlying Web user behaviors. Based on the analysis of these differences, we propose a number of user behavior features to separate Web spam from ordinary pages.

4. USER BEHAVIOR FEATURES OF WEB SPAM PAGES

In this section, we propose six user behavior features that can separate spam pages from ordinary ones. The first five features are from user behavior patterns, and the last feature is a link analysis feature extracted from a user browsing graph that is also constructed with users' Web access log data. We compared the feature distributions of spam and ordinary pages instead of spam and non-spam pages. The reason is that there are huge differences in user visiting patterns between non-spam pages. For example, both CNN homepage and a CNN news article page can be regarded as non-spam while their numbers of user visits/clicks are significantly different. If we constructed a non-spam training set, it would be almost impossible to cover all types of non-spam pages with several hundreds, thousands or even tens of thousands samples. Therefore, all unlabeled Web pages are employed as the "ordinary page set" in feature analysis and algorithm training processes.

In addition to those in our previous work [Liu et al. 2008a][Liu et al. 2008b], three new features named Query Diversity (QD), Spam Query Number (SQN) and User-oriented TrustRank are presented. These features are employed to make the proposed detection framework more effective. Fewer features were used than in the methods proposed in other works; for example, 298 features were adopted in the study by

Agichtein et al. [2006]. This difference is because our framework focused on high-level features, and each of these features was examined to determine whether it was suitable for spam detection. The small number of features makes it possible to gain high performance with relatively simple and efficient learning algorithms, which may be more applicable for the practical Web search environment.

In this section, these features' statistical distributions in the training set will be compared with those in the ordinary page set. The ordinary page set contains all pages in the Web access logs described in Section 3.

4.1 Search Engine Oriented Visit Rate

People visit Web pages through various ways: they may get a recommendation for a Web site from friends or trusted ads, they may revisit valuable pages in their browsers' bookmark or history lists, and they may also follow certain Web pages' out-links according to their interest.

Spam pages try to attract a Web user's attention, but their content is not valuable for most search users. Therefore, few people will get a recommendation for a spam page from a friend, save it in their bookmark lists, or visit it by following a non-spam page's hyperlinks. For most Web spam pages4, a large proportion of their user visits come from search result lists. However, if an ordinary page contains useful information, there are other ways for it to be visited (a person's or Web page's recommendation) other than a search result list.

We define the Search Engine Oriented Visit Rate (SEOV rate) of a certain page p as:

$$SEOV(p) = \frac{\#(Search\ engine\ oriented\ visits\ of\ p)}{\#(Visits\ of\ p)} \quad (1)$$

Web spam pages are seldom visited except through search result lists, but ordinary pages may be visited by other means. Therefore, the SEOV values of Web spam pages should be higher than those of ordinary pages. Our statistical results in Figure 2 validate this assumption.

[4] For some particular types of Web spam pages, such as comment spam on forums/blogs/microblogs, a major part of their user visits may come from other sources (e.g. ordinary forum/blog/microblog posts, content recommendation services, etc.) than search engines.

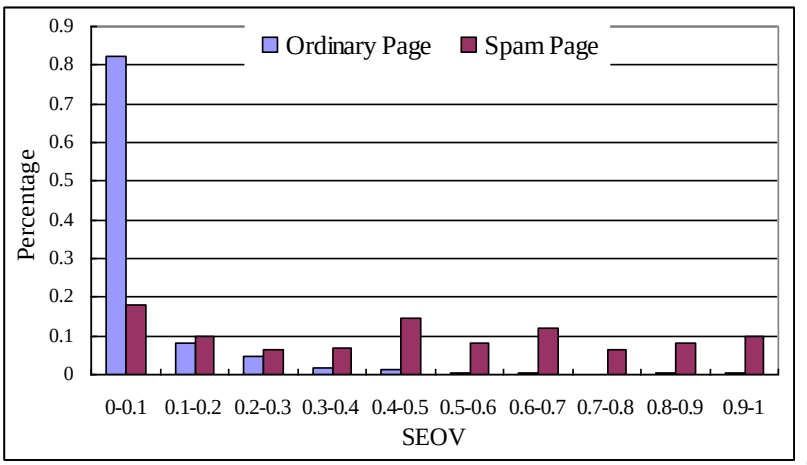


Fig. 2. Search Engine Oriented Visiting (SEOV) distribution of ordinary pages and Web spam pages. (Category axis: SEOV value interval; Value axis: percentage of pages with corresponding SEOV values.)

In Figure 2, the statistics of ordinary pages are shown for all Web sites in the access log described in Section 3. The statistics show that 82% of ordinary pages get less than 10% of their visits from search engines, while almost 60% of Web spam pages receive more than 40% of their visits from search result lists. Furthermore, less than 1% of ordinary Web pages have SEOV values greater than 0.7, while over 20% of spam pages have SEOV values greater than 0.7. Therefore, we can see that most Web spam pages have SEOV values that are higher than ordinary pages because search engines are the target of Web spamming and are sometimes the only way in which spam sites are visited.

4.2 Source Page Rate

Once a hyperlink is clicked, the URLs of both the source page and the destination page are recorded in the Web access log. Each page may appear either as a source page or as a destination page. However, we found that Web spam pages are rarely recorded as source pages. Although spam pages may contain hundreds or even thousands of hyperlinks, most of these links are rarely clicked by most Web users.

We can define the Source Page (SP) rate of a given Web page p as the number of appearances of p as a source page divided by the number of appearances of p in the Web access logs:

$$SP(p) = \frac{\#(p \text{ appears as the source page})}{\#(p \text{ appears in the Web access logs})} \quad (2)$$

The experimental results in Figure 3 show the SP distribution of ordinary pages and spam pages. We can see that most ordinary pages' SP values are larger than those of spam pages. Almost half of the spam pages in the training set rarely appear as the source

page (SP < 0.05). Only 7.7% of spam pages' SP rates are greater than 0.40, while for ordinary pages, the percentage is greater than 53%.

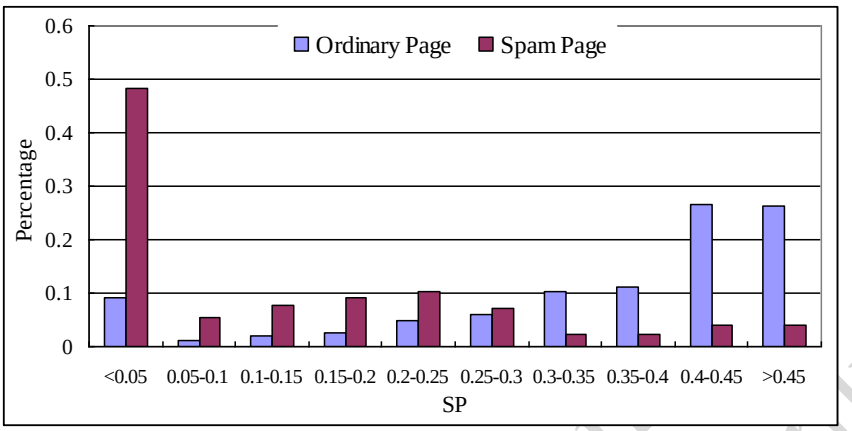


Fig. 3. Source Page (SP) distribution of ordinary pages and Web spam pages. (Category axis: SP value interval; Value axis: percentage of pages with corresponding SP values.)

As in Figure 2, the statistics of ordinary pages are collected for all Web sites in the access log mentioned in Section 3. The differences in the SP value distributions can be explained by the fact that spam pages are usually designed to show users misleading advertisements or low-quality services at the first look. Therefore, most Web users will not click the hyperlinks on spam pages as soon as they notice the spamming activities. Few spam pages appear as source pages because when users visit these pages via hyperlinks, they will end their navigation and follow hyperlinks on other pages.

4.3 Short-time Navigation Rate

User attention is one of the most important resources for Web information providers. Improving the number of user visits and page visits is essential for most commercial Web sites. Therefore, ordinary Web site owners want to keep users navigating within their sites for as long as possible.

However, things are different for Web spammers. Instead of retaining users in their web sites, spammers' major purpose in constructing Web spam sites is to guide users to advertisements or services they do not like to see. They do not expect Web users to navigate inside their Web sites; therefore, when users visit any page in a spam site, advertisements or services are usually shown to them immediately. Meanwhile, this spamming activity causes most Web users to end their navigation in spam sites at once because they do not expect to see such content. Therefore, we can assume that most Web users do not visit many pages inside spam Web sites. We define the Short-time

Navigation rate (SN rate) of a web site to describe this assumption. The SN rate of a given Web site s is defined as:

$$SN(s) = \frac{\#(Sessions\ in\ which\ users\ visit\ less\ than\ N\ \ pages\ in\ s)}{\#(Sessions\ in\ which\ users\ visit\ s)} \quad (3)$$

In contrast to SEOV and SP, SN is a site-based feature to identify Web spamming techniques. The threshold N in its definition is set to 3 based on experience gained in our research.

Most Web users will not continue their visits inside a spam site, but many of them may visit a number of pages in ordinary Web sites because these sites are designed to keep users inside them. Therefore, SN rates of Web spam sites should be much higher than those of ordinary Web sites. The statistical results shown in Figure 4 validate this assumption.

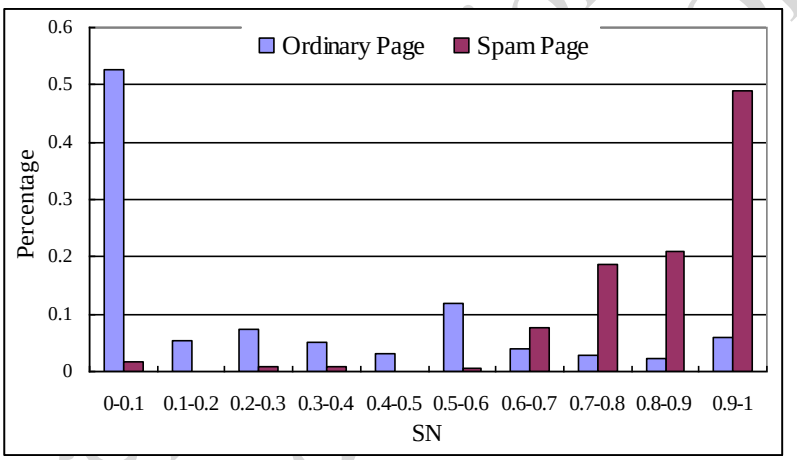


Fig. 4. Short-time Navigation (SN) distribution of ordinary Web sites and Web spam sites. (Category axis: SN value interval; Value axis: percentage of Web sites with corresponding SN values.)

In Figure 4, 53% of ordinary Web sites have SN values less than 0.1, which indicates that over 90% of their visiting sessions contain more than 2 page visits (as mentioned before, N is set to 3 in our SN definition). However, only 14% of the Web spam sites have SN values less than 0.1. Meanwhile, 35% of Web spam sites have SN values greater than 0.80, with users visiting only 1 or 2 pages before leave the site. Therefore, we can see that most Web spam pages' SN values are higher than ordinary pages because they cannot and have no intention of keeping users in their sites.

4.4 Query Diversity

In Section 4.1, we found that most spam pages' user visits are directed by search engines. When we analyzed the queries that lead to these spam pages, we found that features can

be derived from these queries to separate ordinary and spam pages. The two features proposed in Section 4.4 and 4.5 are based on this analysis.

To attract attention from more Web users, many Web spam pages try to become referred by search engines for varies kinds of queries. For example, a spam page may contain keywords from various topics (ring tone download, software download, mobile phone usage FAQs, and so on) so that it will be retrieved by different queries that might describe totally different topics and user intentions. On the contrary, ordinary pages tend to contain a relatively small number of topics, and therefore the number of search query topics that lead to them is relatively small. To describe this difference between spam and ordinary pages, we propose the feature Query Diversity (QTD) to measure the diversity of a page's query topics.

We define the Query Diversity of a certain page p as:

$$QD(p) = Number\ of\ Query\ Topics\ that\ lead\ to\ user\ visit\ for\ p \tag{4}$$

To calculate QD(p), we should obtain the number of query topics that lead to p. With the Web access log data, we collected all the queries from which p's visits were referred. We then grouped these queries into query topics according to their term similarity.

To calculate the similarity between queries, we define the content-based similarity function of certain queries a and b as:

$$Similarity(a,b) = \frac{\#common\ term(a,b)}{Min(length(a), length(b))} \tag{5}$$

Here, length(a) and length(b) refer to the numbers of terms in corresponding queries, and #common term(a, b) is the number of common terms in these two queries. If Similarity(a, b) is greater than a threshold T, a and b are considered to describe a same topic. In our experiment, T was set to 0.2 based on our experience5. The QD values of pages in the training set were calculated, and the statistical distributions are shown in Figure 5.

[5] According to analysis of search queries in our data set, there are 3.11 terms for each query on average after Chinese word segmentation, which is slightly longer than that of English search queries (2.4 terms according to http://en.wikipedia.org/wiki/Web_search_query). Therefore, we believed that the *T* parameter should be similar or a bit higher than that of in Chinese Web environment.

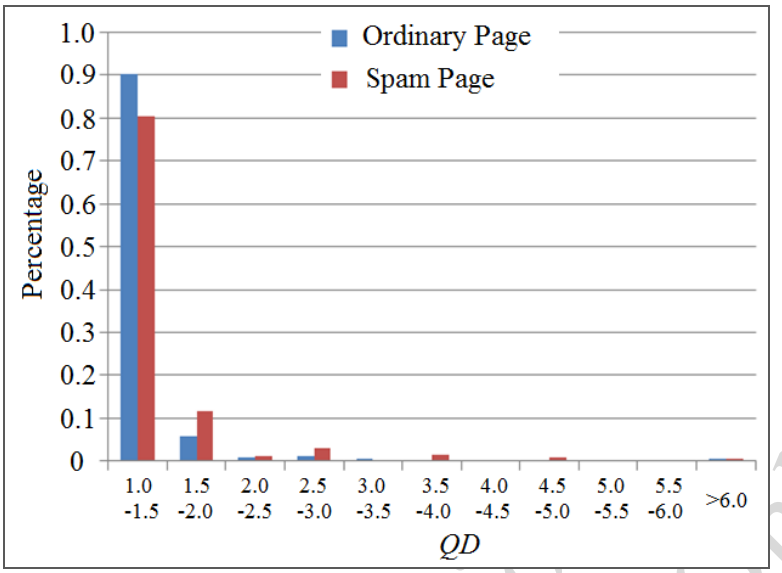


Fig. 5. Query Diversity (QD) distribution of ordinary pages and Web spam pages. (Category axis: QD value interval; Value axis: percentage of pages with corresponding QD values.)

Figure 5 shows that the QD distributions of ordinary and spam pages are different. The QD values of approximately 90% of ordinary pages and 80% of spam pages are less than or equal to 1.5. The percentage of spam pages with QD values greater than 4.0 is 2.1%, which is 2.4 times as much as the corresponding percentage (0.86%) of regular pages. Some ordinary pages also have high QD values; most of these pages are hub pages or Web sites' entry pages. However, in general, a Web spam page's QD value is higher than that of an ordinary page because spam tends to contain multiple content topics to be retrieved by various kinds of query topics. The distributions of QD feature for ordinary and spam pages show that it may not be so effective as SEOV, SP or SN. However, we found that it does help identify some important kinds of spam pages (such as keyword dumping).

4.5 Spam Query Number

When we examined the queries that lead to a large number of Web spam pages, we found that most of these "spam oriented" queries are both popular among users and have relatively few matching resources. If we use P(visit) to represent the probability of a spam page S being visited, then:

$$P(visit) = \sum_i P(query_i)P(visit \mid query_i) \tag{6}$$

In this equation, $query_i$ are the queries that lead to S. Therefore, $P(query_i)$ is the probability that $query_i$ is proposed to a search engine, and $P(visit|query_i)$ is the probability

of visiting *S* while searching for information with $query_i$. From this equation, we can see that spam pages are designed for these kinds of queries because (1) these queries are frequently proposed so that $P(query_i)$ is high and (2) a relatively small number of resources can be retrieved for these queries so that $P(visit|query_i)$ is relatively high.

Therefore, we can see that there are some queries (terms) that are preferred by spammers in designing spam pages, which can be called spam queries (terms). If these queries (terms) can be identified, we can use them to detect possible spam pages based on whether/how many spam queries (terms) direct to these pages.

To identify these queries (terms), we first collected queries that led to 2,732 spam Web sites. These spam sites were annotated by assessors from a commercial search engine company while examining random sampled search results in October, 2008. Word segmentation and stop word removing were performed, and the terms that appeared in more than 4 Web sites' query sets were identified as spam terms. We collected a total of 2,794 spam terms in this process. Spam terms that appear the most frequently are shown in the following table.

Table II. Ten most frequently appeared spam query terms

Spam query term (in Chinese)	English Translation and explanations	Number of spam sites returned as a search result to queries containing the term
图片	Photos (usually appeared together with pornography terms)	1127
五月天	May day (a famous Chinese adult Web site which is already banned)	803
人体	Human body	673
小说	Novel	582
艺术	Art (usually appeared together with pornography terms)	515
电影	Movie	498
免费	Free	484
欧美	Western (usually appeared together with pornography terms)	483
美女	Beauty	475
视频	video	452

From Table II we can see that most of the frequently-appeared spam terms are related with pornography resources, which are illegal in China. These resources are requested by a large number of users while few of them are available. This makes them preferred by spammers.

To evaluate how many spam terms were associated with certain Web pages, we defined the spam query number (SQN) of a certain Web page p as:

$$SQN(p) = Number\ of\ \ Spam\ Query\ Terms\ that\ lead\ \ to\ user\ visit\ \ for\ \ p \tag{7}$$

From the definition of SQN, we can see that a spam term list is required to decide whether one query term is spam term or not. This list can be obtained either with an automatic method or a manual one. In Equation (6), we pointed out that spammers prefer query terms that are frequently proposed yet with few reliable Web sources. Therefore, an automatic method may be developed based on both query frequency and corresponding resource number. The features and algorithm proposed in Castillo et al. [2008] or Ntoulas et al. [2006] may also be employed to find spam terms. However, as described above, we choose a manual method that may produce more credible spam terms than automatic methods. This method is based on the fact that each commercial search engine heavily relies on result annotation and performance evaluation to improve ranking algorithms. In result annotation process, assessors make relevance judgment for result documents. Usually, they also identify low quality or spam results in this process and the annotated spam pages can be employed as a good source for spam term generation. Therefore, it is usually convenient for commercial search engines to keep an up-to-date spam term list for the calculation of SQN.

Spam query terms frequently appear in queries that lead to spam pages, while they rarely appear in the ordinary pages' corresponding queries. Therefore, we assumed that the SQN values of Web spam pages should be higher than those of ordinary pages. The statistics in Figure 6 validate this assumption.

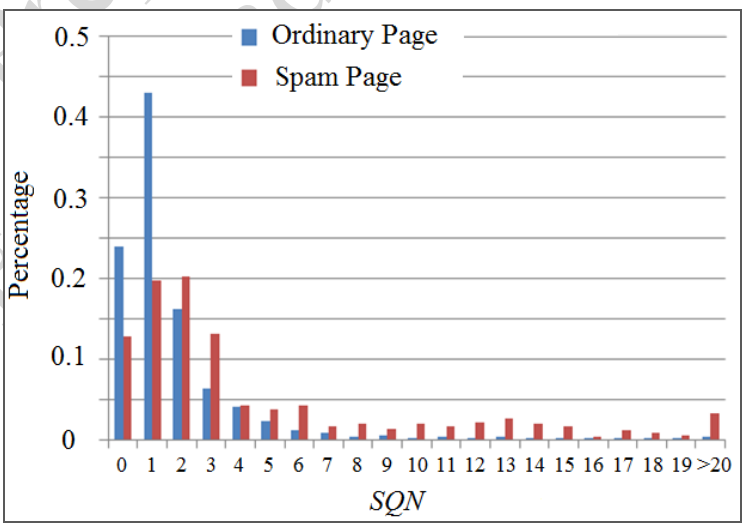


Fig. 6. Spam query term (SQN) distribution of ordinary pages and Web spam pages. (Category axis: SQN value interval; Value axis: percentage of pages with corresponding SQN values.)

In Figure 6, 83% of ordinary pages' SQN values are less than or equal to 3.0, while only half of the Web spam pages' SQN values are less than or equal to 3.0. Furthermore, less than 2% of ordinary Web pages have SQN values greater than 10.0, while over 20% of spam pages' SQN values are greater than 10.0. Therefore, we can see that Web spam pages' SQN values are greater than those of ordinary pages because Web spam pages frequently use spam query terms to gain a higher ranking.

4.6 User-oriented TrustRank

TrustRank is an effective link analysis algorithm that assigns trust scores to Web pages and is usually adopted to identify spam pages. However, just as in other hyperlink analysis algorithms, the TrustRank algorithm is based on two basic assumptions [31]: the recommendation assumption and the topic locality assumption. It is assumed that if two pages are connected by a hyperlink, the linked page is recommended by the page that links to it (recommendation) and the two pages share a similar topic (locality). However, the Web is filled with spam and advertising links, so the assumptions and the original TrustRank algorithm have many problems in the current Web environment.

To obtain a better page quality estimation result with current link analysis algorithms, researchers proposed many techniques such as advertisement detection, page segmentation [Cai et al. 2004] and page block importance identification [Song et al. 2004]. However, most of these methods involved page content or HTML structure analysis, which may not be quite efficient for a great number of Web pages. In order to solve this problem, researchers such as Liu et al. [2008] ran link analysis algorithms on the user browsing graph instead of the entire hyperlink graph. It is believed that the user browsing graph can avoid many problems appearing in the practical Web because links in the browsing graph are actually chosen and clicked by users. In our previous work [Liu et al. 2009], we found that the size of the user browsing graph edge set was only 7.59% of the original hyperlink graph edge set while the two graphs shared a same vertex set. Although it means that a large part of hyperlink information are reduced, we found that the performance of the TrustRank algorithm on the user browsing graph had stable improvement in either high quality page or spam page selection.

Because the performance improvement is stable and the computation cost is significantly reduced (size of the graph is much smaller), we employed the TrustRank algorithm on the user browsing graph and called these results user-oriented TrustRank scores. With the Web access logs described in Section 3, we constructed a user browsing graph and compared the performance of the TrustRank algorithm on different link graphs.

For the TrustRank algorithm, a high quality page "seed" set should be constructed. In our experiments, we followed the construction method proposed by Gyöngyi et al. [2004], which is based on an inverse-PageRank algorithm and human annotation. Altogether, 1,153 high-quality Web sites were selected as the seed set for the TrustRank algorithm, and the iteration time was set to 20 considering the size of our browsing graph. For the whole hyperlink graph, we adopted the data set described in Liu et al. [2009], which contains over 3 billion pages (all the pages in a commercial search engine's index).

We constructed a pairwise orderedness test set to evaluate the performance. The method of the pairwise orderedness test was first introduced by Gyöngyi et al. [2004] at the same time that the TrustRank algorithm was proposed. We constructed a pairwise orderedness test set composed of 700 pairs of Web sites. These pairs were annotated by the same assessors who helped us construct the spam training set described in Section 3. It is believed that pairwise orderedness shows the differences in reputation and user preference for a pair of sites. The experimental results of the user-oriented TrustRank and the traditional TrustRank are shown in Table III.

Table III. Pairwise orderedness accuracy for the TrustRank algorithm on different graphs

Graph	**Pairwise Orderedness Accuracy**
User browsing Graph	0.9586
Hyperlink Graph	0.8571

From the experimental results in Table III, we can see that the user browsing graph has better performance based on the metric of pairwise orderedness accuracy. This result agrees with the results of [Liu et al. 2008][Liu et al. 2009], who found that the link analysis algorithm can better represent users' preferences when performed on the user browsing graph than on hyperlink graph. Liu et al. [2008] based their conclusion on a comparison of the top 20 results ranked by different link analysis algorithms. Our results were obtained with a much larger pairwise test set and are therefore more reliable.

5. USER BEHAVIOR BASED SPAM DETECTION ALGORITHM

To combine the user-behavior features described in Section 4, we used a learning-based mechanism to finish the Web spam detection task. Web spam detection has been viewed as a classification problem in many previous works such as Svore et al. [2007]. Web spam page classification shares a similar difficulty with the Web page classification problem described by Yu et al. [2004] in the lack of negative examples. Positive examples (Web spam pages) can be annotated by a number of assessors using techniques such as pooling [Voorhees 2001]. However, there are so many negative pages that a

uniform sampling without bias is almost impossible because it is regarded as a challenge for Web researchers [Henzinger 2003]. In order to avoid the uniform sampling and annotation processes of a huge number of negative examples, researchers developed a number of algorithms to learn from large scale unlabeled data.

Several learning mechanisms have been proposed for Web page classification based on unlabeled data and a number of positive examples. Techniques such as the PEBL learning framework (based on two-class SVM learning) [Yu et al. 2004], semi-supervised learning [Nigam et al. 2000], single-class learning [Denis 1998], and one class SVM (OSVM) [Manevitz and Yousef 2002] have been adopted to solve the problem. Unlike these algorithms, our anti-spam approach is based on the naïve Bayesian learning framework, which is believed to be among the most competitive algorithms for practical learning problems [Mitchell 1997]. We adopt Bayesian learning because it is both effective and efficient for the problem of learning to classify documents or Web pages. It can also provide explicit probabilities of whether a Web page is a spam page, which can potentially be adopted in result ranking of search engines. Experiment results in Section 6.3 also show its effectiveness via comparison with OSVM, two-class SVM, and decision tree learning algorithms.

For the problem of Web spam classification, we consider two cases: the case in which classification is based on only one feature and the case in which multiple features are involved.

Case 1: Single feature analysis. If we adopt only one user-behavior feature A, the probability of a web page p with feature A being a Web spam can be denoted by:

$$P(p \in Spam \mid p \ has \ \ feature \ A) \tag{8}$$

We can use Bayes theorem to rewrite this expression as:

$$P(p \in Spam \mid p \ has \ \ feature \ A) = \frac{P(p \ has \ \ feature \ A \mid p \in Spam)}{P(p \ has \ \ feature \ A)} \times P(p \in Spam) \tag{9}$$

In Equation (9), $P(p \in Spam)$ is the proportion of spam pages in the entire page set. This proportion is difficult to estimate in many cases, including our problem of Web spam page classification. However, if we just compare the values of $P(p \in Spam \mid p \ has \ \ feature \ A)$ in a given Web corpus, $P(p \in Spam)$ can be regarded as a constant value and would not affect the comparative results. In a fixed corpus, we can rewrite equation (8) as:

$$P(p \in Spam \mid p \ has \ \ feature \ A) \propto \frac{P(p \ has \ \ feature \ A \mid p \in Spam)}{P(p \ has \ \ feature \ A)} \tag{10}$$

Considering the terms in Equation (10), $P(p\ has\ feature\ A \mid p \in Spam)$ can be estimated using the proportion of A-featured pages in the Web spam page set, while $P(p\ has\ feature\ A)$ equals the proportion of pages with feature A in a given corpus. Here we obtain:

$$\frac{P(p\ has\ feature\ A \mid p \in Spam)}{P(p\ has\ feature\ A)} = \frac{\#(p\ has\ feature\ A \cap p \in Spam)}{\#(Spam)} \Big/ \frac{\#(p\ has\ feature\ A)}{\#(CORPUS)} \tag{11}$$

If the sampling of Web spam pages can be regarded as an approximately uniform process (in contrast to the task of sampling non-spam Web pages uniformly, it is a much easier task because spam pages are supposed to share similar user behavior features), we can rewrite the numerator of (11) as:

$$\frac{\#(p\ has\ feature\ A \cap p \in Spam)}{\#(Spam)} = \frac{\#(p\ has\ feature\ A \cap p \in Spam\ sample\ set)}{\#(Spam\ sample\ set)} \tag{12}$$

Substituting expressions (11) and (12) into (10), we obtain:

$$P(p \in Spam \mid p\ has\ feature\ A) \propto \frac{\#(p\ has\ feature\ A \cap p \in Spam\ sample\ set)}{\#(Spam\ sample\ set)} \Big/ \frac{\#(p\ has\ feature\ A)}{\#(CORPUS)} \tag{13}$$

We can see that in (13) #(Spam sample set) and #(CORPUS) can be estimated by the sizes of the training set and the corpus. #(P has feature A $\cap$ p$\in$Spam sample set) and #(P has feature A) can be obtained by the number of pages with feature A in both the training set and the corpus. Therefore, all terms in (13) can be obtained by statistical analysis of a Web page corpus, we can calculate the probability of being a Web spam for each page according to this equation.

Case 2: Multiple feature analysis. If we use more than one feature to identify Web spam pages, the naïve Bayes theorem assumes that the following equation holds:

$$P(p\ has\ feature\ A_1, A_2, ..., A_n \mid p \in Spam) = \prod_{i=1}^{n} P(p\ has\ feature\ A_i \mid p \in Spam) \tag{14}$$

For the problem of page classification with user-behavior features, we further found that the following equation also approximately holds according to Table IV.

$$P(p\ has\ feature\ A_1, A_2, ..., A_n) = \prod_{i=1}^{n} P(p\ has\ feature\ A_i) \tag{15}$$

Table IV. Correlation values between user-behavior features of Web pages

	SEOV	***SP***	***SN***	***TrustRank***	***QD***	***SQN***

SEOV	1.0000					
SP	0.0255	1.0000				
SN	0.1196	0.1221	1.0000			
TrustRank	-0.0027	0.0163	0.0444	1.0000		
QD	0.0506	0.0512	0.0776	0.3447	1.0000	
SQN	0.1460	0.0706	0.1747	0.1186	0.5712	1.0000

The correlation values in Table IV show that most of the features are approximately independent of each other because their correlation values are relatively low. This may be explained by the fact that these features were obtained from different information sources and thus have little chance of affecting one another.

One exception is that QD and SQN are not independent because they are both extracted from the queries that lead to Web pages. Therefore, in the following parts of the paper, we will retain the SQN feature and discard the QD feature to validate the independence between user behavior features. When we only consider the other five features listed in Table IV, we can see that their attribute values are independent and conditionally independent given the target value.

From the statistical analysis in Table IV, the following equations approximately hold for the Web spam page classification task according to the naïve Bayes assumption:

$$\begin{aligned} & P(p \in Spam \mid p \text{ has feature } A_1, A_2, ..., A_n) \\ &= \frac{P(p \text{ has feature } A_1, A_2, ..., A_n \mid p \in Spam) P(p \in Spam)}{P(p \text{ has feature } A_1, A_2, ..., A_n)} \\ &\approx P(p \in Spam) \prod_{i=1}^{n} \frac{P(p \text{ has feature } A_i \mid p \in Spam)}{P(p \text{ has feature } A_i)} \\ &= P(p \in Spam)^{1-n} \cdot \prod_{i=1}^{n} \frac{P(p \text{ has feature } A_i \mid p \in Spam) P(p \in Spam)}{P(p \text{ has feature } A_i)} \\ &= P(p \in Spam)^{1-n} \cdot \prod_{i=1}^{n} P(p \in Spam \mid p \text{ has feature } A_i) \\ &\propto \prod_{i=1}^{n} P(p \in Spam \mid p \text{ has feature } A_i) \end{aligned} \tag{16}$$

If we substitute (9) into (12), we obtain the following equation for multi-feature cases:

$$\begin{aligned} & P(p \in Spam \mid p \text{ has feature } A_1, A_2, ..., A_n) \\ &\propto \prod_{i=1}^{n} \left(\frac{\#(p \text{ has feature } A_i \cap p \in Spam \text{ sample set})}{\#(Spam \text{ sample set})} \Big/ \frac{\#(p \text{ has feature } A_i)}{\#(CORPUS)} \right) \end{aligned} \tag{17}$$

According to this equation, the probability of a web page being a Web spam page can be calculated with information from the Web corpus and its corresponding spam page sample set. Therefore, it is possible to use the following algorithm to accomplish the spam identification task.

Algorithm 1. Web spam detection with user behavior analysis

1. Collect Web access log (with information shown in Table I) and construct access log corpus *S*;
2. Calculate *SEOV*, *SP*, and *SQN* scores according to Equations (1), (2), (4), and (7) for each Web page in *S*;
3. Calculate *SEOV*, *SP*, and *SQN* scores for each Web site in *S* by averaging the scores of all pages in the site;
4. Calculate *SN* score for each Web site in *S* according to Equation (3);
5. Construct user browsing graph with S and calculate user-oriented TrustRank scores according to the algorithm proposed in Gyöngyi et al [2004].
6. Calculate $P(Spam \mid SEOV, SP, SN, SQN, TrustRank)$ according to Equation (13) for each Web page in *S*.

After performing **Algorithm 1** on *S*, we obtain a spam probability score for each Web page in *S*. This score can be used to separate spam pages from ordinary ones.

6. EXPERIMENTS AND DISCUSSIONS

6.1 Experiment Setups

After Bayesian learning on the training set, we constructed a classifier that could assign probabilities of being Web spam based on user behavior analysis. We then used this classifier to assign spam probability values for all Web sites recorded in the Web access log data.

We adopted a site-level method instead of a page-level method to avoid the data sparseness problem because when we adopted the page-level method, we found that a large number of Web pages are visited only a few times; this makes the calculation of some user-behavior features (such as SEOV and SP) unreliable. For example, a certain outdated news page may not be interesting for most users. When a certain user searches for related information, it is possible for him to visit this page via a search result page. This may be the only user visit during a period of time and the SEOV value for this page would be 1.00, which indicates a possible spam page. However, integrating information from all pages within a site can avoid such problems because there are some possibly up-to-date news pages in the same site.

The construction of both the training set and the test set are described in Section 3.4. All of the experimental results shown in this section are based on these data sets.

6.2 Spam Detection Performance of User-behavior-oriented Method

After annotation, we chose ROC curves and corresponding AUC values to evaluate the performance of our spam detection algorithm. This is a useful technique for organizing classifiers and visualizing their performance and has been adopted by many other Web spam detection studies such as Web Spam Challenge[6] and [Svore et al. 2007][Abernethy et al. 2008]. The AUC values of the detection algorithm are shown in Table V.

Table V. Correlation values between user-behavior features of Web pages

Feature Selection	**AUC value**	**Performance loss**
All features	0.9150	/
All features except for *SEOV*	0.8935	-2.40%
All features except for *SP*	0.9010	-1.55%
All features except for *SN*	0.8872	-3.13%
All features except for user-oriented *TrustRank*	0.8051	-13.64%
All features except for *SQN*	0.8831	-3.61%
User-oriented *TrustRank* only	0.8128	-12.57%

We can see from Table V that with all features proposed, the detection method had an AUC value of 0.9150. This indicates that our detection method has a probability of 0.9150 to rank a spam page before an ordinary page. Table V also shows that dropping any proposed feature will hurt the performance. The performance loss caused by dropping a certain feature can be regarded as a metric for this feature's spam detection capacity. Therefore, user-oriented TrustRank is the most effective feature because performance will be reduced the most if we discard this feature. We can also see that with all five proposed features, the performance will be the best.

Another interesting finding is that if we remove one of the four features besides user-oriented TrustRank, performance loss will not be as great (less than 5%). However, when we just utilized user-oriented TrustRank to finish the task of spam detection, we found that the performance is not as good (AUC = 0.8128). This result is better than the results obtained with all features except for user-oriented TrustRank but is worse than the performance with all features (from 0.9150 to 0.8128, with 12.57% performance loss). Therefore, although user-oriented TrustRank is the most effective feature, the spam detection performance comes from a combined effort.

6.3 Comparison with other Learning Algorithms

[6] http://webspam.lip6.fr/

To prove the effectiveness of our proposed learning-based detection algorithm (Algorithm 1), we compared the performance of this algorithm to some other learning based methods. Algorithm 1, support vector machine, and decision tree algorithms were adopted to combine the five features proposed in Section 4 (except *QD* because its correlation value with *SQN* is relatively high).

The libSVM[7] and C4.5[8] toolkits were adopted in our comparison experiments as implementations of the SVM and decision tree algorithms. For SVM learning, we used both one-class SVM and two-class SVM to finish the spam identification task. Two-class SVM training is based on both spam and non-spam samples while one-class SVM only depends on spam samples. There are only spam samples in our training set. Therefore, we prepare an extra non-spam training set (containing 904 Web sites) for training two-class SVM. It was constructed with the same method as the spam test set described in Section 3.3 but the size was smaller than the test set so that the number of spam and non-spam samples was similar.

We compare the performance of our detection algorithm (Algorithm 1) with one-class SVM because both of them do not require non-spam training samples (although Algorithm 1 also needs unlabeled data in the training process). For two-class SVM learning, we used C-SVC SVM and chose the radial basis function (RBF) as the kernel function. For both one-class SVM and two-class SVM, five-fold cross validation and grid search were employed for tuning the parameters of C, gamma and nu.

For the decision tree learning process, it also requires both spam and non-spam samples in the training process. Therefore, both the spam training set and the constructed non-spam training set were employed. After C4.5 learning, the original tree was composed of 23 nodes while the pruned tree contains 11 nodes.

Performance evaluation results of the learning algorithms are shown in Table VI and Figure 7. We can see that the precision values for one-class SVM (when recall equals to 25%, 50%, and 75%) and decision tree (when recall equals to 75%) are not included. This is because recall for these algorithms do not reach those values in our experiments.

Table VI. AUC/precision-recall comparison of different learning algorithms in spam detection

	Precision			**AUC**	**AUC Compared with Algorithm 1**
	Recall=25%	**Recall=50%**	**Recall=75%**		

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[8] http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html

SVM(two class)	100.00%	65.34%	25.57%	0.8815	-2.39%
SVM(one class)	/	/	/	0.5072	-77.96%
Decision Tree	64.03%	50.50%	/	0.7149	-26.26%
Algorithm 1	100.00%	76.14%	43.75%	0.9150	/

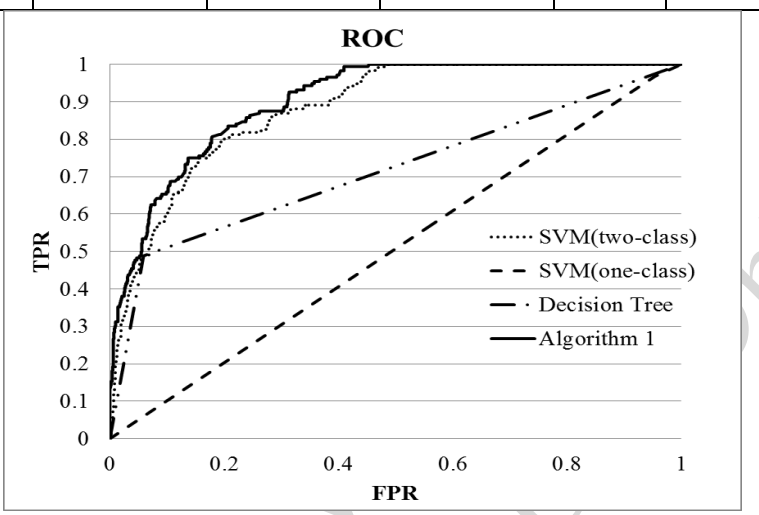


Fig. 7. ROC curves of different learning algorithms in spam detection

From the experimental results in Table VI and Figure 7, we found that with the metric of AUC values, Algorithm 1 (the proposed naïve Bayes based algorithm) outperforms the other algorithms. The ordinary two-class SVM algorithm was the second best while one-class SVM performed the worst. With precision-recall evaluation metrics, the proposed algorithm gained the highest precision scores when recall equaled to 25%, 50% and 75%. Another phenomenon was that performance of the proposed algorithm was even better than the other algorithms while the recall value was relatively high (75%). It showed that the proposed algorithm which relied on both spam and unlabeled data can separate a large part of spam sites from other Web sites.

The phenomena that Algorithm 1 performed better than other algorithms was possibly caused by the fact that Bayes learning is effective especially when the correlation between features is small (see Table IV). However, we believe that this result is highly correlated with the fact that Algorithm 1 utilizes information from the unlabeled data, whose size is much greater than the spam/non-spam training set. One class SVM performed the worst (slightly better than random results) because it only adopted information from the positive examples in the training set and misses other useful information. Two-class SVM and decision tree did not perform so well because although

they employed non-spam training samples, the training samples could only cover a small proportion of non-spam sites on the Web.

6.4 Comparison with Existing Spam Detection Algorithms

As stated in Section 2, many spam fighting techniques have been proposed to detect Web spam pages and reduce them from search engine results. Among these algorithms, some were not designed for specific types of spam such as the content-based method proposed by Cormack et al. [2011] and TrustRank algorithm proposed by Gyöngyi et al. [2004]. We compare the performance of the user-behavior-oriented algorithm with these two methods because the proposed one is also possible to detect various types of spam (see Section 6.5).

The content-based method [Cormack et al. 2011] was selected because it proved to be effective on both TREC Web track and WEBSPAM Challenge benchmarks. In order to obtain information required by the algorithm, we employed the SogouT corpus which contains over 130 million pages crawled in the middle of 2008 in Chinese Web. Pages in the constructed spam/non-spam training set and the test set were filtered from the corpus and overlapping byte 4-gram features were extracted. Because the Web access log data was collected a few months later than the SogouT corpus, some Web sites in the training/test set were not included in the corpus. We found that 641, 634 and 1,377 sites were retained for the spam training set, non-spam training set and test set, respectively. We exactly followed the algorithm implementation and parameter settings of the content-based method except that each Chinese character was treated as two bytes. In this way, letter-based 4-gram features in the original algorithm were replaced with character-based 2-gram features in our implementation (contents without Chinese characters remained to be represented with 4-gram features).

For the TrustRank algorithm, we adopted the same seed set employed by the user-oriented TrustRank feature described in Section 4.6. TrustRank was performed with default parameters (decay factor = 0.85, number of iterations = 20)on the whole hyperlink graph described in Liu et al. [2009], which contains over 3 billion pages (all the pages in a commercial search engine's index).

After performing the content-based and link-based algorithms on the test set, we evaluated their spam detection performance with the metric of AUC and precision-recall. Results are shown in Table VII and Figure 8.

Table VII. AUC/precision-recall comparison of different spam detection algorithms

	Precision	AUC

	Recall = 25.00%	Recall = 50.00%	Recall = 75.00%	
Content-based algorithm [Cormack et al. 2011]	81.63%	7.65%	4.08%	0.6414
Link-based algorithm [Gyöngyi et al. 2004]	74.43%	34.09%	18.75%	0.7512
User-behavior-based algorithm (Algorithm 1)	100.00%	76.14%	43.75%	0.9150

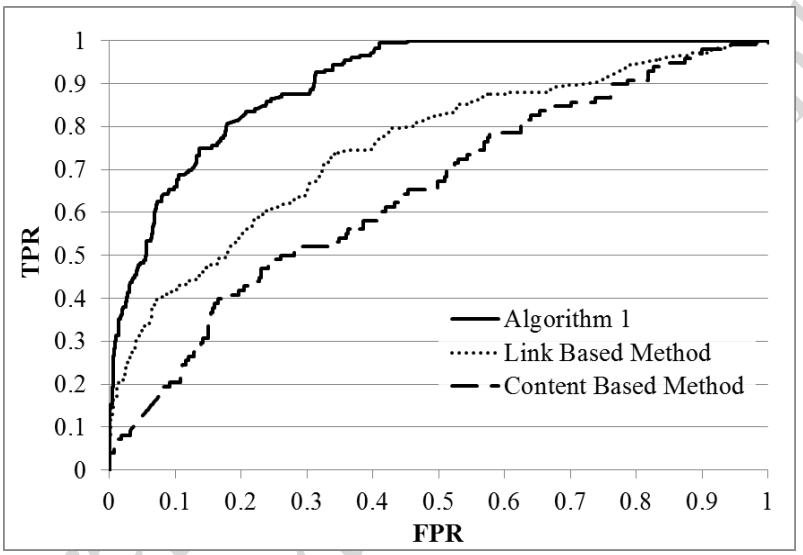


Fig. 8. ROC curves of different spam detection algorithms

Experimental results in Table VII and Figure 8 show that the proposed user-behavior-oriented algorithm outperforms both content-based and link-based algorithms with either AUC or precision-recall metrics. TrustRank gained higher AUC value than the content-based method but its precision value is lower when recall is 25%. It means that a large part of top-ranked pages in the spam result list given by the content-based method were actually spam pages.

When we look into the content-based spam detection results, we found that this method could identify most spam that focused on honeypot topics already existing in the spam tra ining set. However, for spam topics not in the training set, its detection performance was not so good. Although the size of the training set was smaller for the content-based method due to the pages missing from the SogouT corpus, we believe that this wasn't the key reason for its relatively low performance. Constructing a training set

with all possible spam topics would be labor-consuming and it would also be rather difficult to keep it up-to-date. Compared with the content-based algorithms, the proposed user-behavior-oriented algorithm does not require page content information and the involvement of huge scale content-based features. Although it may miss some content information recorded on Web pages, user behavior features such as SQN can be employed to introduce spam topic information in the detection process. Therefore, the user-behavior-based detection performance was better than the content-based method.

Another finding from Table VII and Figure 8 is that the TrustRank algorithm performed worse than the proposed user-behavior-oriented algorithm. Its AUC performance (0.7512) was lower than the user-oriented TrustRank feature (0.8128) proposed in Section 4.6. It accords with our findings in Liu et al. [2009] that hyperlink analysis algorithm performed better on user browsing graph than on the whole hyperlink graph.

While looking into the identification results, we also found that the proposed algorithm could identify some new types of Web spam pages that existing algorithms could not. Figure 8 shows a spam page that was detected by the user behavior based algorithm while ignored by both TrustRank [Gyöngyi et al. 2004] and the content-based filtering methods proposed by [Cormack et al. 2011].

Differently from traditional content spamming pages, the spam pages shown in Figure 8 used search results from the largest Chinese search engine (Baidu.com) as its contents. The search results were crawled from Baidu.com with a hot "honeypot" query (the name of a TV show) and employed to cheat search engines. With this spamming technique, spammers put result snippets (Figure 9(a), 9(b)) and/or search result links (Figure 9(a)) on the pages. Although they seemed to be relevant to the popular TV show, they actually mean nothing for Web users because most users had just visited the search result pages. Spammers designed such pages to increase user visits of their Web sites while both search engines and users were misled to a useless resource.

With the help of search result contents, this kind of spam page was more difficult to detect than other content spamming pages such as the ones using repetition, dumping, weaving or stitching techniques 0. Firstly, the honeypot terms (queries) appear many times in search results because search engines try to show users keyword matching in results. Secondly, the honeypot keywords appear together with their contexts in result snippets. Due to these two reasons, this kind of spam page appears to be a highly relevant page for the honeypot query. Therefore, traditional content-based spam features such as document length [Ntoulas et al. 2006], title length [Ntoulas et al. 2006], N-gram language

model [Ntoulas et al. 2006][Cormack et al. 2011] and POS tags [Piskorski et al. 2008] may not be able to detect this type of spam pages.

However, things are different for the user-behavior-oriented detection framework. Spamming techniques employed by spammers do not change the fact that spam pages are designed to cheat Web users instead of providing reliable information or resources. Therefore, user behavior features (especially SEOV, SN and SQN) can tell that these pages are spam pages.

(a)

(b)

Fig. 9. Two spam pages that used search results of a popular honeypot query “珠光宝气” (The Gem of Life, a popular TV show) as its content. (a: http://hi.baidu.com/yuehe/blog/item/f2c16081acaa26dfbd3e1e41.html; b: http://sms1000.blog.hexun.com/27098683_d.html)

6.5 Detection of Various Kinds of Spam Pages

One problem with the state-of-the-art anti-spam techniques is that they cannot be adopted to detect various kinds of spam pages. Therefore, we wanted to examine whether our algorithm was able to solve this problem. According to the experimental results in Table VIII, we found that term-based, link-based, and other kinds of spamming techniques can all be detected by the proposed user behavior based algorithm.

Table VIII. Page types of the top 300 possible spam pages identified by our spam detection method

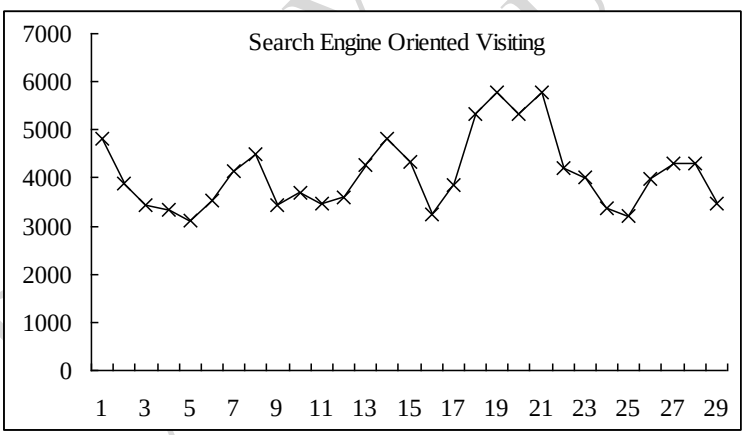
Page Type	Percentage
Non-spam pages	5.33%
Web spam pages (Term spamming)	31.67%
Web spam pages (Link spamming)	25.33%
Web spam pages (Term + Link spamming)	11.33%
Web spam pages (Other spamming)	27.00%
Pages that cannot be accessed	9.33%

In Table VIII, 300 pages that were identified as spam by our algorithm are annotated with their page types. These pages are the top-ranked pages in the possible spam list ranked by spam probabilities. First, we found that most of the identified pages were spam pages, while 5.33% of these pages were not spam pages. However, further analysis into these non-spam pages showed that they were mostly low-quality pages that adopted some kind of SEO technique to attract users. Second, there were also a number of pages that could not be accessed at the time of assessment. We believe that most of these pages were previously spam because spam pages usually change their URL to bypass search engines’ spam list. Meanwhile, ordinary pages would not change their domain name because doing so hurts their rankings in search engines. Finally, we can see that both term-based and link-based spamming techniques can be identified by our algorithm. We adopted user behavior features to detect Web spam, which made it possible to identify Web spam independent of spamming technique types. This can be regarded as a possible solution to the “multi-type problem” proposed in Section 2.2.

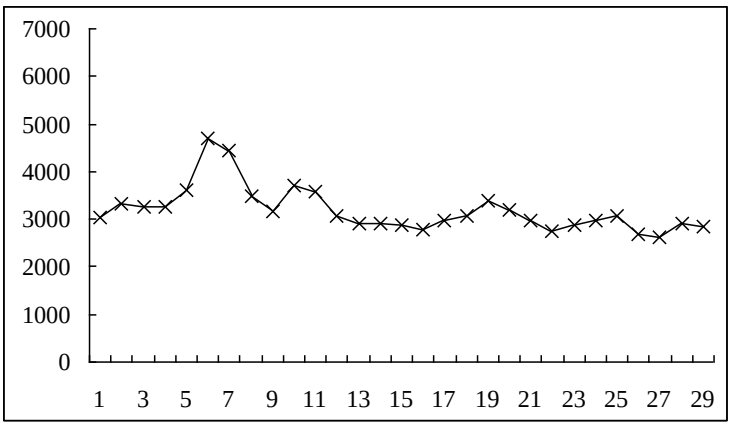
6.6 Detection of Newly-appeared Spam Pages

We mentioned the "timeliness problem" in Section 2.2 and regarded it as one of the most challenging problems for current anti-spam techniques. We found that our algorithm could identify various kinds of Web spam pages, and we wanted these spam pages to be identified as soon as possible. Therefore, we designed the following experiments to see whether our algorithm could identify Web spam more quickly than the spam detection methods adopted by commercial search engines.

In Table VIII, we obtained the page type information for the top 300 pages in our spam probability list. Among these pages, 256 of them were various types of Web spam, and 28 could not be accessed at the time of annotation. The Web access log data we adopted were collected from Nov. 12th, 2008 to Dec. 15th, 2008. Therefore, these spam sites were detected by our algorithm on Dec. 15th, 2008. If search engines can detect these spam sites as quickly as our algorithm, search engine-oriented visiting to these sites should be reduced. To determine how the amount of search engine-oriented visiting evolves over time, we collected the next month's Web access data (from Dec. 22nd, 2008 to Jan. 20th, 2009) for these spam sites and extracted the search engine oriented-visiting data. Figure 10 shows the results.



(a)



(b)

Fig. 9. Search engine oriented visiting for spam sites detected by the proposed algorithm. (a): from Nov.12nd, 2008 to Dec. 15th, 2008; (b): from Dec. 22nd, 2008 to Jan. 20th, 2009

In Figure 10, five widely used Chinese search engines (Baidu, Yahoo! China, Google China, Sogou, and Yodao) were adopted to collect the search oriented user-visiting data. We can see from Figure 10(a) that during the time period when the user behavior data were collected, the number of search engine oriented visiting was relatively high (4085 UVs per day, on average). In the next month, the average visiting dropped to approximately 3189 UVs per day. This result indicates that the search engines identified some of the spam sites and stopped placing them in their search result lists. However, the total amount of search engine oriented visiting was still approximately 80% of that of the first 30 days. The proposed spam detection method can identify these spam sites at the end of the first 30 days (Dec. 15th, 2008); meanwhile, search engines still lead users to these spam sites even after approximately 40 days. This means that our spam detection method is able to detect newly-appeared Web spam pages, and the detection is faster than the anti-spam techniques adopted by commercial search engines.

6.7 Size of Usage Data

With the experimental results shown above, we found that the proposed user-behavior-based framework is effective for spam detection and especially for identifying newly-appeared spam and various types of spam. However, there are also some limitations of the proposed method. Some of the features adopted in our framework are derived from the users' search engine interaction behavior, such as SEOV, QD, and SQN. This fact indicates that one needs to wait for the spam content to be indexed by a search engine and served to users for several weeks before spam can be eliminated from the search index.

In the process of identifying spam with the wisdom of the crowds, spam pages are exposed to some Web users. These users will be affected by the existence of these spam sites, and then the algorithm can identify spam based on user behavior patterns. Although we believe that the proposed framework can eventually identify Web spam, the number of users that are affected by spam should be as small as possible.

In order to find out how many users would be affected before spam pages were identified by our algorithm, we examined how much usage data is needed for the proposed spam detection framework to perform effectively. For all 1,997 Web sites in the constructed test set (described in Section 3.3), we calculated their UV (user visit) statistics from Nov. 12th, 2008 to Dec. 15th, 2008. We found that the most frequently visited site is www.tianya.com, which was visited by 2,376,743 users. There are also several sites with low UVs, however, all sites were visited at least by 10 users because the data cleansing process described in Section 3.2 reduced those with fewer 10 UVs. After we got the UV statistics, we segmented the Web sites in test set into 10 buckets. Each bucket contained approximately 200 samples. Size of these buckets and their corresponding UV ranges are shown in Table IX.

Table IX. Size of the segmented buckets and corresponding UV ranges for the test set

Bucket No.	Number of Web sites	UV range
1	196	10-15
2	196	16-35
3	197	36-100
4	202	100-300
5	201	301-1,500
6	202	1,501-5,100
7	201	5,101-16,000
8	205	16,001-53,000
9	205	53,001-250,000
10	192	250,001-
10	192	250,001-

After segmenting Web sites in test set into these buckets, we examine the spam detection performance in each bucket by the metric of AUC values. Experimental results are shown in Figure 10. From these results, we got the following findings:

Firstly, spam detection performance of all buckets were above 0.76 by the metric of AUC values, it means that the proposed algorithm was effective (more effective than the performances of both content-based and link-based methods according to the statistics in Table VII) even with relatively small amount of usage data. Secondly, the user behavior

based algorithm generally performed better with more usage data because spam detection performances of bucket 6-10 were significantly better than those of bucket 1-5. However, while the number of user visits increases from 10-15 (Bucket No. 1) to 301-1,500 (Bucket No. 5), detection performance remained almost the same. It means that detection performance for Web sites with less than 1,500 UVs (during the 33 days when we collected usage data, about 45 UVs per day) may not be as effective as those with more UVs. Compared with the large number of search engine users, the required data size is a relatively small amount usage data, especially when spam pages are designed to attract many users' attention. Therefore, we believe that the proposed algorithm can help commercial search engines in their spam detection process.

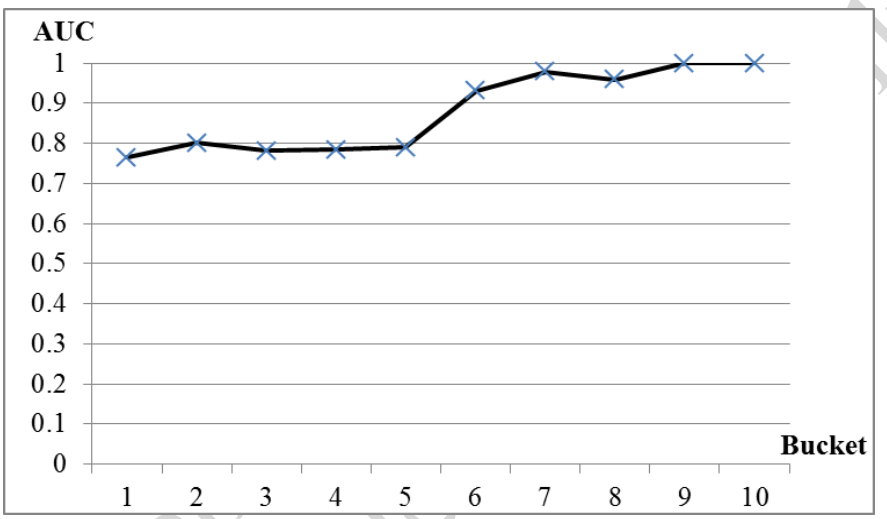


Fig. 10. Spam detection performance for Web sites in different bucket

However, the proposed user-behavior spam detection framework should not be regarded as a totally different alternative to current spam detection methods. This method can be employed to detect newly-appeared spam sites and, more importantly, new spam techniques but it cannot identify spam pages with little usage data. For commercial search engines, many spam pages exist in their index data and only a tiny part of them can be ranked high in a result list and shown to users due to various reasons. For the spam pages not shown to users, it is almost impossible for our method to identify them because little user behavior data could be collected. However, reducing these pages may be quite important for improve system efficiency of search engine systems. Therefore, our algorithm should be used together with traditional spam detection methods that focus on stopping spam as soon as it appears on the Web instead of when they affect users.

7. CONCLUSIONS AND FUTURE WORK

Most spam detection approaches focus on predefined types of spam pages using content or hyperlink analysis. In contrast to these traditional methods, we propose a user-behavior-oriented Web spam detection algorithm. This algorithm analyzes large-scale Web access logs and exploits the differences between Web spam pages and ordinary pages in user behavior patterns. We combined machine learning techniques and descriptive analysis of user-behavior features of Web spam pages. In this way, we come to a better and deeper understanding of the relationship between user visiting patterns and Web page spamming activities.

Currently, the user-behavior-oriented approach may not be as effective as state-of-the-art anti-spam algorithms in identifying certain types of Web spam. However, with the help of Web user behavior, the proposed method can detect various kinds of spam pages no matter what spamming techniques they adopt. Newly-appeared spam can also be identified as soon as a relatively small number of users are affected. This method may not replace existing anti-spam algorithms, but it can help search engines find the most bothersome spam types and be aware of newly-appeared spam techniques.

In the near future, we hope to extend this framework to embody page content and hyperlink features. We also plan to work on a Web page quality estimation model for Web information management tools based on the findings in this paper.

ACKNOWLEDGMENTS

In the early stages of this work, we benefited enormously from discussions with Yijiang Jin, Zijian Tong, Kuo Zhang and Jianli Ni; we thank Xiaochuan Wang from Sogou.com for kindly offering help in corpus construction and annotation; we also thank the anonymous referees of this paper, for their valuable comments and suggestions.

REFERENCES **<<ENTRIES ARE ALPHABETICAL BY LAST NAME OF PRIMARY AU>>**

ABERNETHY J., CHAPELLE O. AND CASTILLO C. 2008. WITCH: A New Approach to Web Spam Detection. *Yahoo! Research Report*. No. YR-2008-001

AGICHTEIN E., BRILL E., AND DUMAIS S. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '06). ACM, New York, NY, USA, 19-26.

AMITAY, E., CARMEL, D., DARLOW, A., LEMPEL, R., AND SOFFER, A. 2003. The connectivity sonar: detecting site functionality by structural patterns. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia* (Nottingham, UK, August 26 - 30, 2003). ACM, New York, NY, USA, 38-47.

BACARELLA V., GIANNOTTI F., NANNI M., AND PEDRESCHI D., 2004. Discovery of ads Web hosts through traffic data analysis. In *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (DMKD), New York, NY, USA: ACM. 76–81.

BECCHETTI L., CASTILLO C., DONATO D., LEONARDI S., AND BAEZA-YATES R.. Using Rank Propagation and Probabilistic Counting for Link Based Spam Detection. In Proceedings of the Workshop on Web Mining and Web Usage Analysis.

BILENKO, M. AND WHITE, R. W. 2008. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *Proceeding of the 17th international Conference on World Wide Web.* ACM Press, New York, NY, 51-60.

BRIN, S. AND PAGE, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the Seventh international Conference on World Wide Web 7* (Brisbane, Australia). 107-117.

BUEHRER G., STOKES J.W., AND CHELLAPILLA K. 2008. A large-scale study of automated web search traffic. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web* (AIRWeb '08), Carlos Castillo, Kumar Chellapilla, and Dennis Fetterly (Eds.). ACM, New York, NY, USA, 1-8.

CAI, D., YU, S., WEN, J., AND MA, W. 2004. Block-based web search. In *Proceedings of the 27th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Sheffield, United Kingdom, July 25 - 29, 2004). SIGIR '04. ACM Press, New York, NY, 456-463.

CASTILLO C. AND DAVISON B. 2011. Adversarial Web Search. *Foundations and Trends in Information Retrieval*. 4, 5 (May 2011), 377-486.

CASTILLO C., CORSI C., DONATO D., FERRAGINA P. AND GIONIS A. Query-log mining for detecting spam. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web* (AIRWeb '08), Carlos Castillo, Kumar Chellapilla, and Dennis Fetterly (Eds.). ACM, New York, NY, USA, 17-20.

Chellapilla K. and Chickering D. M. 2006. Improving cloaking detection using search query popularity and monetizability. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web* (AIRWeb), 17–24.

CNNIC (CHINA INTERNET NETWORK INFORMATION CENTER). 2009. *Search engine user behavior research report*.

CRASWELL, N., HAWKING, D., AND ROBERTSON, S. (2001). Effective site finding using link anchor information. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '01). ACM, New York, NY, USA, 250-257.

DAVISON B. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, AAAI Press, July 2000. Presented at the AAAI-2000 workshop on Artificial Intelligence for Web Search, Technical Report WS-00-01. 23-28.

DENIS, F. PAC Learning from Positive Statistical Queries. 1998. Proceedings of the 9th international Conference on Algorithmic Learning theory. *Lecture Notes In Computer Science*, vol. 1501. London: Springer-Verlag. 112-126.

FETTERLY, D., MANASSE, M. AND NAJORK, M. 2004. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases.* S. Amer-Yahia and L. Gravano, editors, 1–6.

FUXMAN, A., TSAPARAS, P., ACHAN, K., AND AGRAWAL, R. 2008. Using the wisdom of the crowds for keyword generation. In *Proceeding of the 17th international Conference on World Wide Web*. ACM Press, New York, NY, 61-70.

GENG, G., WANG, C., LI, Q., XU, L., AND JIN, X. 2007. Boosting the Performance of Web Spam Detection with Ensemble Under-Sampling Classification. In *Proceedings of the Fourth international Conference on Fuzzy Systems and Knowledge Discovery* (FSKD 2007) Vol.4 - Volume 04 (August 24 - 27, 2007). FSKD. IEEE Computer Society, Washington, DC, 583-587.

GORDON V. CORMACK, MARK D. SMUCKER, CHARLES L. A. CLARKE. 2011. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *Information Retrieval*. 1-25.

GYONGYI, Z. AND GARCIA-MOLINA, H. 2005. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan. 1-9.

GYÖNGYI, Z., GARCIA-MOLINA, H., AND PEDERSEN, J. 2004. Combating web spam with trustrank. In *Proceedings of the Thirtieth international Conference on Very Large Data Bases* - Volume 30. 576-587.

HENZINGER, M.R., MOTWANI, R., SILVERSTEIN, C. 2003. Challenges in Web Search Engines. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (2003) 1573-1579.

JANSEN J.B. 2007. Click fraud, *Computer*, vol. 40, no. 7, pp. 85–86,

KLEINBERG. J.M. 1999. Authoritative sources in a hyperlinked environment. 1999. *Journal of the ACM, 46(5)*:604-632.

KRISHNAN, V. AND RAJ, R. Web Spam Detection with Anti-Trust-Rank. 2006. In *proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web* (AIRWeb), August 2006.

LIU Y., CEN R., ZHANG M. MA S., RU L. 2008a. Identifying Web Spam with User Behavior Analysis. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web* (AIRWeb '08), Carlos Castillo, Kumar Chellapilla, and Dennis Fetterly (Eds.). ACM, New York, NY, USA,

LIU, Y., GAO, B., LIU, T., ZHANG, Y., MA, Z., HE, S., AND LI, H. 2008. BrowseRank: letting web users vote for page importance. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '08). ACM, New York, NY, USA, 451-458.

LIU Y., ZHANG M. MA S., RU L. 2008b. User behavior oriented web spam detection. In *Proceeding of the 17th international Conference on World Wide Web* (Beijing, China, April 21 - 25, 2008). WWW '08. ACM, New York, NY, 1039-1040.

LIU, Y., ZHANG, M., MA, S., RU, L. 2009. User Browsing Graph: Structure, Evolution and Application. Late Breaking result session of *the 2nd ACM International Conference on Web Search and Data Mining* (WSDM 2009).

MANEVITZ, L. M. AND YOUSEF, M. 2002. One-class SVMs for document classification. *Machine Learning*. Res. 2: 139-154.

MITCHELL, T. 1997. Chapter 6: Bayesian Learning, in Mitchell, T., *Machine Learning*, McGraw-Hill Education.

NIGAM, K., MCCALLUM, A. K., THRUN, S., AND MITCHELL, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*. 39(2-3): 103-134.

NTOULAS, A., NAJORK, M., MANASSE, M., AND FETTERLY, D. 2006. Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on World Wide Web* (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 83-92.

PISKORSKI J., SYDOW M., AND WEISS D. 2008. Exploring linguistic features for Web spam detection: A preliminary study. In *Proceedings of the Fourth International Workshop on Adversarial Information Retrieval on the Web* (AIRWeb), New York, NY, USA: ACM. 25–28.

SILVERSTEIN, C., MARAIS, H., HENZINGER, M., AND MORICZ, M. 1999. Analysis of a very large web search engine query log. *SIGIR Forum 33, 1 (Sep. 1999)*, 6-12.

SONG, R., LIU, H., WEN, J., AND MA, W. 2004. Learning block importance models for web pages. In *Proceedings of the 13th international Conference on World Wide Web* (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM Press, New York, NY, 203-211.

SULLIVAN D. 2006. Searches Per Day. Retrieved from *search engine watch web site* http://searchenginewatch.com/reports/article.php/2156461.

SVORE, K., WU, Q., BURGES, C. AND RAMAN, A. 2007. Improving Web Spam Classification using Rank-time Features. In *proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb '07)*, May 2007.

VOORHEES. E. M. 2001. The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems* (CLEF '01), Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck (Eds.). Springer-Verlag, London, UK, 355-370.

WANG Y., MA M., NIU Y., AND CHEN H. 2007. Spam double-funnel: connecting web spammers with advertisers. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 291-300.

WU, B. AND DAVISON, B. Cloaking and redirection: a preliminary study. 2005. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan.

YU, H., HAN, J., AND CHANG, K. C. 2004. PEBL: Web Page Classification without Negative Examples. *IEEE Transactions on Knowledge and Data Engineering* 16, 1 (Jan. 2004), 70-81.

Received November 2009; revised March 2011; accepted June 2011.