Identifying Spam Link Generators for Monitoring Emerging Web Spam

Young-joo Chung
chung@tkl.iis.u-tokyo.ac.jp

Masashi Toyoda
toyoda@tkl.iis.u-tokyo.ac.jp

Masaru Kitsuregawa
kitsure@tkl.iis.u-tokyo.ac.jp

Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo, 153-8505, JAPAN

ABSTRACT

In this paper, we address the question of how we can identify hosts that will generate links to web spam. Detecting such spam link generators is important because almost all new spam links are created by them. By monitoring spam link generators, we can quickly find emerging web spam that can be used for updating existing spam filters. In order to classify spam link generators, we investigate various link-based features including modified PageRank scores based on white and spam seeds, and these scores of neighboring hosts. An online learning algorithm is used to handle large scale data, and the effectiveness of various features is examined. Experiments on three yearly archives of Japanese Web show that we can predict spam link generators with a reasonable performance.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

Link analysis, Web spam, Information retrieval

1. INTRODUCTION

As the Web plays an important role in economy, social activities, and information sharing, search engines become indispensable tools to access the huge amount of information. Considering about the half of users look at no more than top five results in a search result list [1], it is clear that a higher ranking in the result list brings more traffic and profits to web sites. As a result, many web sites started using unfair ways, so called *web spamming*, to boost their rankings in the list.

WICOW'10, April 27, 2010, Raleigh, North Carolina, USA.

Spammers use various techniques that manipulate textual contents and link structure of web sites. They insert popular keywords into their pages, and copy relative documents from other sites to make their sites look useful. They also create densely connected link structure, for optimizing link-based ranking scores such as PageRank [2].

Detecting web spam is a challenging task because new sophisticated techniques have been continuously invented to evade existing anti-spam techniques. For example, recent spammers copy short text segments from various related sites for avoiding document copy detection techniques. Existing spam detection techniques mainly use machine learning techniques, and they perform very well on benchmarks [3]. However, they need to update their classifier for adapting to newly appeared spamming techniques.

Our goal is monitoring emerging spam hosts, so that we can quickly adapt to new spamming techniques. Since new spam hosts might not be detected by existing spam filters, we need to identify where spam hosts are created in the Web. For this purpose, we focus on hosts that frequently generate outgoing links to spam hosts. We call such hosts *spam link generators*. Both normal and spam hosts can be spam link generators. Spammers can create links on normal hosts using link hijacking techniques, such as posting spam comments on blogs, and buying expired domains that are linked by normal hosts. Hosting service companies also point to spam hosts if their customers create spam pages in them.

By observing our temporal Japanese Web archive, we found that the number of spam link generators is relatively small compared with the total number of hosts while they produce almost all new spam links. If we can identify spam link generators, the cost for observing emerging spam would be drastically reduced.

In this paper, we try to identify spam link generators using the online learning algorithm with various link-based features. Our work can contribute in following situations:

- By observing spam link generators, we can promptly collect samples of new spam hosts. If those hosts use new spamming techniques, we can use those samples as training sets for updating existing spam classifiers.
- When normal hosts are detected as spam link generator, we can notify their web masters that those hosts are vulnerable to spammers. The web masters can examine that reasons, and make it resilient against spammers.
- Search engines can penalize spam links from spam link

generators for improving their link-based ranking, and can reduce crawling priority of spam links. Detailed analysis inside spam link generators is necessary to determine which pages or document object model(DOM) nodes should be penalized.

A binary classifier is used to identify spam link generators. Given a host, our classifier predicts whether the host is a spam link generator or not. A spam link generator is defined as the host that will generate spam links more than some threshold in a time period.

Link-based features including a PageRank score and the number of links are examined to implement the classifier. In addition, information of whether an out-neighboring host is spam or not is necessary to identify spam link generators. Trustworthiness of a host itself would be also related to the increase of spam links. To measure the trustworthiness of a host, we compute white and spam scores using two different modified PageRank algorithms. The white score is obtained using white seed hosts while the spam score is calculated using spam seed hosts. A host is considered as a normal one if it has a high white score and a low spam score, and vice versa. With these features, we train our classifier using an online learning algorithm that is more suitable for web-scale learning problems [4].

The rest of the paper is organized as follows. In Section 2, we review previous studies related with PageRank and spam detection. Section 3 describes our method for identifying spam link generators in the Web in detail. In Section 4, the experimental results are presented. We summarize and conclude our work in Section 5.

2. RELATED WORK

PageRank [2] is a link-based ranking algorithm that models a random surfer. In PageRank, the surfer move to new pages by either following a outgoing link or jumping randomly to pages that are not linked by a current page. Thus, PageRank can be considered as the probability with which a surfer stays at that page. Since high PageRank score of a page implies that page is more likely to be accessed by web surfers, spammers try to boost PageRank scores using link spamming techniques like creating a link farm [12]. A link farm consists of a target page and boosting pages. All boosting pages link to the target page in order to increase its rank score. Then, the target page distributes its boosted PageRank score back to supporter pages. By this, members of a link farm can boost their PageRank scores.

Several approaches have been suggested for detecting and demoting link spamming. To understand the characteristics of spamming, Gyöngyi et al. described various web spamming techniques in [11]. Fetterly et al. found that outliers in statistical distributions are very likely to be spam by analyzing statistical properties of linkage, URL, host resolutions and contents of pages [10].

To demote spam pages and make PageRank resilient to link spamming, Gyöngyi et al. suggested TrustRank [5]. In TrustRank, a web surfer jumps randomly to only pages that are judged good by human expert. By this, good pages will have high TrustRank score while spam pages will not. Optimizing the link structure is another approach to demote link spam. Carvalho et al. proposed the idea of noisy links, a link structure that has a negative impact on link-based ranking algorithms [19]. Qi et al. also estimated the quality of links by similarity of two pages [20].

To detect link spam, Anti-TrustRank is suggested by Krishnan et al. in [13] A web surfer in Anti-TrustRank either follows incoming links or jumps to spam pages selected manually. Consequently, spam pages will have high Anti-TrustRank score. Benczur et al. introduced SpamRank [14]. SpamRank checks PageRank score distributions of all in-neighbors of a target page. If this distribution is abnormal, SpamRank regards a target page as spam and penalizes it. Gyöngyi et al. suggested Mass Estimation in [18]. They evaluated *spam mass*, a measure of how many PageRank scores a page gets through links from spam pages. Graph algorithms are also used to combat with link spamming. Saito et al. decomposed the Web graph into strongly connected components and discovered that large components are spam with high probability. Link farms in the core were extracted by maximal clique enumeration [6]. Our previous work expanded this work by proposing recursive strongly connected component decomposition [22].

Spam detection can be regarded as classification problem by machine learning algorithm. Castillo et al. employed content-based and link-based features to classify web spam in [16]. Spam classification using changes in link structure over time was proposed by [8]. Shen et al. introduced historical link-based feature such as incoming-link growth rate and death rate to train a classifier. Dai et al. employed historical content features to improve the performance of spam classification in [9].

As far as we know, there are few studies focusing on monitoring emerging web spam. In our previous work, we detected a hijacked site that is normal and points to spam sites by link hijacking [7]. We showed that hijacked sites can be detected using scores obtained by modified versions of PageRank algorithm. In this paper, we consider not only normal hosts but spam hosts that point to spam hosts. We also try to predict a host that will generate links to spam hosts.

3. SPAM LINK GENERATOR IDENTIFICATION

In this section, we present the definition of spam link generators and briefly introduce an online learning algorithm used for our experiments. Furthermore, we describe link-based features that are used to identify generators. Notations in Table 1 are used.

3.1 Definition of Spam Link Generator

If a spam host s exist in the Web at time t, there should be a host g in time $t-1$ that generate links to s between $t-1$ and t. The host g can be selected by investigating the difference between the number of spam out-neighbors in time $t-1$ and t. That is, if $\|sOut(g)_t\| > \|sOut(g)_{t-1}\|$, we call the host g a spam link generator (The definition of $sOut(g)$ is described in Section 3.3). Formally, spam link generators are defined as:

$$G = \{g \mid \|sOut(g)_t\| - \|sOut(g)_{t-1}\| \geq \epsilon\},$$

where ϵ is a growth threshold to determine the degree of spam link growth that should be satisfied by spam link generators.

3.2 Learning Algorithm

An online learning algorithm is used to build our classifier. The online learning algorithm is suitable for a large scale data such as Web because it guarantees a fast convergence while achieving similar or even better accuracy than offline learning algorithms such as support vector machine(SVM) [4] [21]. Moreover, since new spamming techniques are invented rapidly, classifiers related with web spamming should be updated frequently. Online algorithms allow an easier update than offline algorithms when obtaining a new spam sample.

In online learning, a classifier tries to assign a correct label on each sample that comes into in sequential manner. We can denote a pair of sample and its label in round t by $(\mathbf{x}_t, y_t)$ where $\mathbf{x}_t$ is a feature vector of a sample and $y_t \in \{+1, -1\}$ is its label. At each round, the algorithm predicts a label of a sample based on its weight vector $\mathbf{w}_t$ and produces $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$ as a *margin*. Such a margin is can be interpreted as the distance between the sample and the hyperplane that divide classes. If the margin is positive, prediction was correct. Otherwise, algorithm modify weigh vector $\mathbf{w}$ to produce more accurate prediction on next coming samples $\mathbf{x}_{t+1}$.

We use Passive-Aggressive(PA) algorithm [17] that tries to update the classification algorithm as little as possible while achieving at least a unit margin on the most recent sample. In other words, PA algorithm updates the weight vector by solving the following optimization problem:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t. } y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq 1.$$

$\mathbf{w}_{t+1}$ remains $\mathbf{w}_t$ whenever the distance between a sample and hyperplane exceed a confidence margin. If not, $\mathbf{w}_{t+1}$ is updated as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t.$$

In PA algorithm, τ_t is defined as:

$$\frac{l_t}{\|\mathbf{x}_t\|^2},$$

where l_t is $\max\{0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$.

Since PA algorithm updates the weight vector as much as possible to close new samples, it can be easily influenced by noisy samples. In order to solve this, PA-I algorithm that allows a gentler update strategy is suggested. In PA-I, τ_t is given by:

$$\min\left\{C, \frac{l_t}{\|\mathbf{x}_t\|^2}\right\},$$

where C is *aggressiveness parameter*. Using small C, we can weaken the effect of noisy samples.

Table 1: Notations for feature definitions.

Notation	Meaning
N	The number of node in Web graph. Node can be a page, host or site.
$In(p)$	The set of nodes pointing to p
$Out(p)$	The set of nodes pointed to by p
$wOut(p)$	The set of normal nodes pointed to by p
$sOut(p)$	The set of spam nodes pointed to by p
S^+	The set of normal seed node
S^-	The set of spam seed node

3.3 Features

PageRank. PageRank [2] computes the importance of each host based on the link structure. The basic idea of PageRank is that a page is important if it is linked by many other important pages. PageRank is defined as following matrix equation:

$$\mathbf{p} = \alpha \cdot \mathbf{T} \times \mathbf{p} + (1 - \alpha) \cdot \mathbf{d}$$

where $\mathbf{p}$ is PageRank score vector, $\mathbf{T}$ is a transition matrix. $T(p, q)$ is $1/\|Out(q)\|$ if there is a link from page q to page p, and 0 otherwise. The decay factor $0 < \alpha < 1$ (usually 0.85) is necessary to guarantee convergence and to limit an effect of rank sink. $\mathbf{d}$ is a uniformly distributed random vector. Instead of following links to next pages, we can jump from a page to a random one chosen according to distribution $\mathbf{d}$. Spammers try to boost the PageRank score of their hosts and to plant links on non-spam hosts with a high PageRank score. As a result, PageRank score can affect the growth of spam links of a host.

White score and Spam score. We use the core-based PageRank algorithm with white and spam seed sets [18] for the white and spam scores calculation. Core-based PageRank assigns initial scores on seed pages that are selected by a human expert. Such scores are propagated through outgoing links during computation. Thus, if we select reputable pages as a seed, good pages will have a high score after computation. On the other hand, if spam seed set is used for score calculation, spam pages will have a high score. A core-based PageRank score vector $\mathbf{p}'$ is given by:

$$\mathbf{p}' = \alpha \cdot \mathbf{T} \times \mathbf{p}' + (1 - \alpha) \cdot \mathbf{d}^{\nu}$$

where a random jump distribution $\mathbf{d}^{\nu}$ is :

$$d_p{}^{\nu} = \begin{cases} 1/N, & \text{if } p \text{ is in seed set } S \\ 0, & \text{otherwise} \end{cases}.$$

Core-based PageRank scores with white seeds($\mathbf{PR}^+$) is used as white scores and Core-based PageRank scores with spam seeds($\mathbf{PR}^-$) is used as spam scores. [1]

Relative Trust. We define *Relative Trust*($\mathbf{RT}$) of each host in order to measure the trustworthiness of a host. A host will be trustworthy only when it has a high white score and a low spam score, and vice versa. Therefore, $\mathbf{RT}$ is the difference between the white and spam scores of a host. $\mathbf{RT}$ is given by:

$$\mathbf{RT}(p) = \log\big(\mathbf{White}(p)\big) - \log\big(\mathbf{Spam}(p)\big) - \delta \ .$$

where $\mathbf{White}(p)$ is a white score of a host p, and $\mathbf{Spam}(p)$ is a spam score of a host p. We used log value since the distribution of core-based PageRank scores obeys power law. If $\mathbf{RT}(p)$ is higher than zero, p is more likely to be a normal

[1]TrustRank [5] and Anti-TrustRank [13] also can be used to calculate the white and spam scores. Unlike core-based PageRank, TrustRank and Anti-TrustRank use the random jump that is biased to small and highly selective seed sets($1/|S^+|$ and $1/|S^-|$). This approach is useful when we try to detect either only normal or spam hosts [5]. In this paper, however, we use white and spam score not for detecting normal or spam hosts but measuring trustworthiness. Since the core-based algorithm uses a more moderate random jump than TrustRank or Anti-TrustRank, white and spam scores could propagate further from seeds.

host. In contrast, if $\mathbf{RT}(p)$ is lower than zero, p is more likely to be spam.

A threshold δ is introduced to reduce the influence caused by the different sizes of seed sets for the white and spam score. Since the core-based PageRank algorithm assigns the initial score only to seed hosts, the total amount of scores for propagation depends on the number of seed hosts. As a result, the average of the white scores and the spam scores will be different if the size of white and spam seed set are significantly different.

We use the δ value obtained by the difference between the average of the initial white scores and that of the spam scores in order to compensate for the size difference of two seed sets.

$$\delta = \log\left(\frac{\|S^+\|}{N}\right) - \log\left(\frac{\|S^-\|}{N}\right),$$

where the first term represents the logarithm of the average of the initial scores of $\mathbf{PR}^+$, and the second term represents that of $\mathbf{PR}^-$. By δ value, we could remove the difference caused by different average of the initial white and spam scores from $\mathbf{RT}$.

Outgoing and Incoming link related measures. Features related with neighboring hosts are also considered. We count the number of links to spam-like hosts and normal-like hosts. We use $\mathbf{RT}$ to determine whether a out-neighboring host is likely to be normal or spam. $wOut$ is the set of out-neighboring hosts of p that are likely to be normal, and $sOut$ is the set of out-neighbors that seem to be spam.

$$wOut(p) = \left\{ w \mid w \in Out(p) \wedge \mathbf{RT}(w) \geq 0 \right\},$$

$$sOut(p) = \left\{ s \mid s \in Out(p) \wedge \mathbf{RT}(s) < 0 \right\}.$$

We will call $wOut$ as normal out-neighbors and $sOut$ as spam out-neighbors of host h. Note that a host with a negative $\mathbf{RT}$ value is not always the spam host. The negative $\mathbf{RT}$ value implies the high likelihood of being spam.

The number of normal links of a host p is $\|wOut(p)\|$, while that of spam link is $\|sOut(p)\|$. In addition, the summation and the mean of $\mathbf{RT}$ of normal and spam out-neighbors are used as features. The $\mathbf{RT}$ sum and the average of normal and spam out-neighbors of a host p are defined as follows:

$$RTSUM_{wOut}(p) = \sum_{w \in wOut(p)} |\mathbf{RT}(w)|,$$

$$RTAVG_{wOut}(p) = \frac{RTSUM_{wOut}(p)}{\|wOut(p)\|},$$

$$RTSUM_{sOut}(p) = \sum_{s \in sOut(p)} |\mathbf{RT}(s)|,$$

$$RTAVG_{sOut}(p) = \frac{RTSUM_{sOut}(p)}{\|sOut(p)\|}.$$

In total, six outgoing-link-related features are obtained. Six features for incoming links are obtained in the same manner.

Hijacked score. The information of how likely a normal host has links to spam hosts can be helpful to identify spam link generators. If a normal host has high probability to be hijacked by spammers, the host would generate spam links, since hijacked hosts tend to be attacked continuously. Based on our previous work [7], we compute a hijacked score that implies how likely a host is hijacked.

First, we create a set H of hijacked candidates. A hijacked host h would be a normal host, and have at least one spam out-neighboring host with a negative $\mathbf{RT}$, a lower white score, and a higher spam score than h.

$$H = \left\{ h \mid \mathbf{RT}(h) \geq 0 \ \wedge \ R(h) \ \neq \phi \right\},$$

where $R(h)$ is:

$$R(h) = \left\{ r \;\middle|\; \begin{array}{l} r \in sOut(h) \ \wedge \\ \mathbf{White}(r) < \mathbf{White}(h) \wedge \\ \mathbf{Spam}(r) > \mathbf{Spam}(h) \end{array} \right\}.$$

Next, we calculate the hijacked score of each hijacked candidate h. The hijacked score of h will be obtained by:

$$\mathbf{H}(h) = \frac{\sum_{w \in wOut(h)} |\mathbf{RT}(w)|}{\|wOut(h)\| + \lambda} \cdot \frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda}.$$

We introduce λ as a smoothing factor to reduce the effect caused by the small number of out-neighbors. Without λ, a host that has small out-neighbors is more likely to obtain a higher hijacked score. This is not desirable because we try to find a host that is hijacked by many spam hosts. To determine λ, we calculate the hijacked scores of 695 labeled sample hosts using different λ values. We change λ from 1 to 101 by adding 10. After hijacked scores are obtained, we manually check top 200 hosts with the high hijacked score whether they are hijacked or not. The λ value that shows the best precision is used to obtain the hijacked scores of whole hosts.

Thus, 17 features are available for the classification. Logarithm of all values except $\mathbf{RT}$ and the hijacked score are taken, and then scaled into [0,1] using their minimum and maximum values. [2]

4. EXPERIMENTS

In this section, we evaluate our approach to spam link generator detection. We describe our data set, and show various features of spam link generators. In addition, we measure the overall performance of our classifier and the effectiveness of each group of link-based features using evaluation metrics.

4.1 Data set and Seed set

Three yearly snapshots of our own Japanese Web archive are used for experiments.[3] These snapshots are built by crawling from 2004 to 2006. Our crawler is based on the breadth first crawling, but it focuses on pages written in Japanese. If a page outside the `.jp` domain is written in Japanese, it also is collected. The crawler stops collecting pages from a site if it is not able to find any Japanese pages on the site within the first few pages. Hence, our snapshot contains pages written in various languages as well as

[2]Note that we exclude the number of spam links that have been generated by a host from the features. This is because we are trying to predict spam link generators when the past data is not available.

[3]Our host graph data set can be distributed to researchers for academic and non-commercial use.

English. Our crawler does not have an explicit spam filter while it detects mirror servers and tries to crawl only representative ones. As a result, our archive includes spam hosts without mirroring.

In this paper, we use host graphs, where each node is a host and each edge between nodes is a hyperlink between pages in different hosts. Host graphs for 2004, 2005 and 2006 are built. In each graph, we include only hosts that exist in the 2006 archive, and do not consider hosts disappeared from 2004 to 2005 since it is difficult to know whether those hosts really disappeared or they were just not reached by our crawler. Consequently, we consider spam link generators that exist for at least one year. The properties of our Web snapshot are shown in Table 2.

Table 2: Properties of data set

Year	2004	2005	2006
Number of nodes(hosts)	2.98M	3.70M	4.02M
Number of edges	67.96M	83.07M	82.08M

To calculate core-based PageRank scores, we construct trust and spam seed sets.

For the white seed set, we compute PageRank score of whole hosts and manually select hosts from 1,000 hosts with a high PageRank score. Well-known hosts like Google, Yahoo!, and MSN, authoritative university and well-supervised company hosts are selected as white seeds. We also add hosts with specific URL including `.gov` (US governmental host) and `.go.jp` (Japanese governmental host) to the trust seed set.

For the spam seed set, we choose hosts with URLs containing spam keywords like `porn,casino,cheap` and `download`, since spammers usually stuff such terms in URLs [10] [16]. In addition, we use spam hosts obtained by the strongly connected component decomposition(SCC) algorithm [6] [22]. A SCC of a graph is a subgraph where every pair of node has a direct path between them. Since spam hosts tend to construct a densely connected link structure, it could be assumed that spam hosts form the SCC. Based on this idea, we decomposed the Web into SCCs and confirmed that 95% of large SCCs around the largest SCC, so called the core, are spam farms in [6]. To find spam farms in the core, we pruned nodes with small degrees from the core, and applied the SCC decomposition algorithm to the pruned core recursively with increasing the degree threshold. Because large SCCs which contains over 100 hosts was very likely to be spam farms [22], we use hosts as spam seeds in large SCCs obtained during nine iterations. Table 3 describes the size of the white and spam seed sets in each year.

Table 3: Size of seed sets in each year

Year	2004	2005	2006
$\|S^+\|$	4,563	5,171	5,183
$\|S^-\|$	306,026	303,851	315,472

4.2 Evaluation Metrics

To evaluate the performance of our classifier, we use precision, recall, and F-measure which are defined as follows:

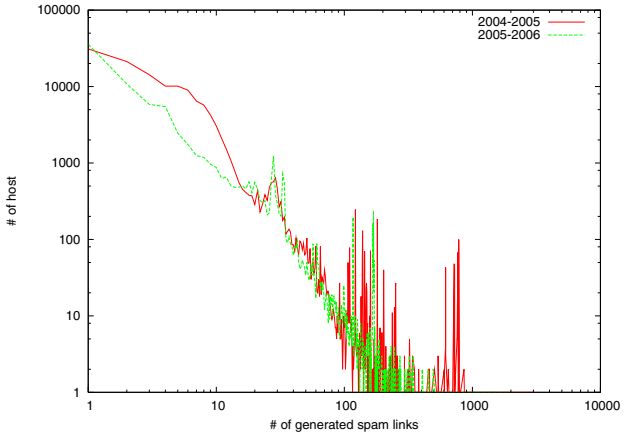

Figure 1: Distribution of the number of host for the generated spam links. Almost all spam links are generated by the half of hosts.

$$\text{Precision} = \frac{|\text{correctly classified spam link generator}|}{|\text{hosts classified as spam link generator}|},$$

$$\text{Recall} = \frac{|\text{correctly classified spam link generator}|}{|\text{spam link generator}|},$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

4.3 Characteristics Of Spam Link Generators

4.3.1 *The number of generated spam links*

To identify spam link generators in the Web, we examine changes in the number of spam links of all hosts and select hosts of which a spam link count increase over the growth threshold as spam link generators. The threshold values 4 and 3 are used for the snapshot of 2004 and 2005, respectively. Table 4 illustrates the number of hosts categorized by changes in their spam link counts.

The proportion of spam link generators is different in each year. In 2004, 8% of hosts generated more than 4 spam links during a year, while about 3% of hosts is selected as a spam

Table 4: Number of hosts categorized by changes in the spam link count

Change in spam link count	2004-2005	2005-2006
Grown	133,268	81,111
Unchanged or Shrunken	752,414	1,107,816
Spam link generator	66,637	34,581

Table 5: Percentage of spam links created by spam link generators to whole spam links.

	2004-2005	2005-2006
Total spam links	1,418,667	898,779
Spam links from generators	1,302,210	841,432
(%)	(91.79%)	(93.62%)

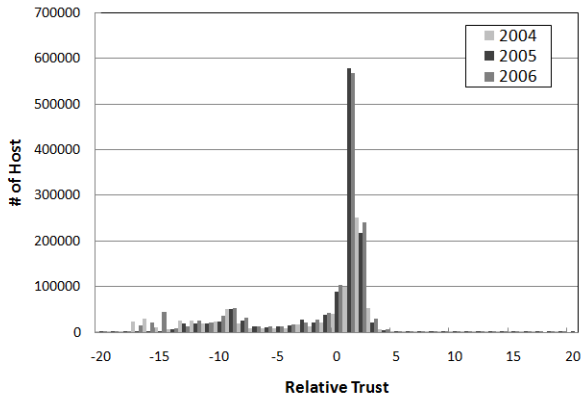

Figure 2: RTs of hosts in 2004, 2005, 2006. Most hosts in every year have RT of +1.

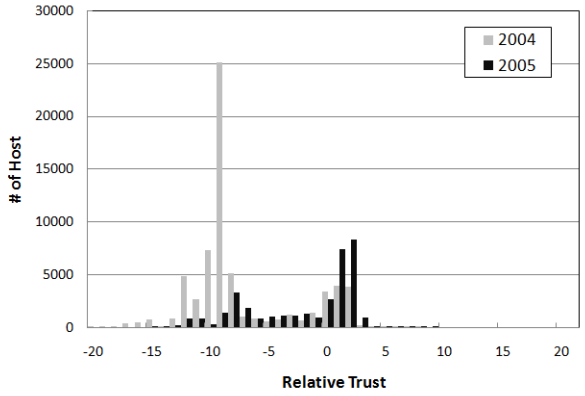

Figure 3: RTs of spam link generators in 2004 and 2005.

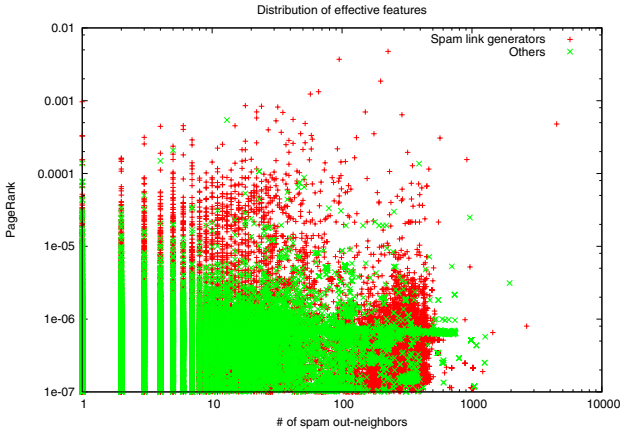

Figure 4: Distribution of the number spam out-neighbors and PageRank in 2004.

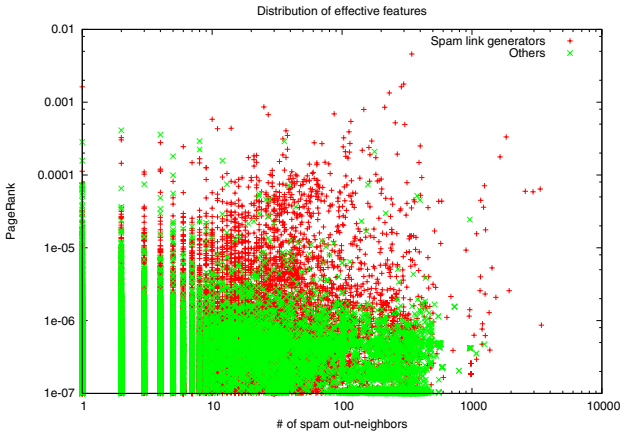

Figure 5: Distribution of the number spam out-neighbors and PageRank in 2005.

link generator in 2005. However, as shown in Figure 1 and Table 5, the percentage of spam links that are created by the spam link generator is similar in both time period. Over 90% of spam links is created by spam link generators. This shows the importance of detecting spam link generators.

We also observed that a considerable number of spam link generators kept their activities for two years. There were about 120 thousand hosts that generated spam links from 2004 to 2005. Among such hosts, about 85 thousand (71%) hosts kept the number of spam links (links might be replaced), and 20 thousand (16%) hosts generated additional spam links between 2005 and 2006.

4.3.2 Relative trust of spam link generators

As depicted in Figure 2 and Figure 3, **RT** distribution of spam link generators differs from that of all hosts. The number of hosts with $\mathbf{RT} > 0$ is greater than that of hosts $\mathbf{RT} < 0$ as shown in Figure 2. 58% of hosts in 2004 and 76% of hosts in 2005 have **RT** value greater than 0. Spam generators tend to have $\mathbf{RT} < 0$ compared to other hosts. The percentage of spam generators with $\mathbf{RT} < 0$ is 83% in 2004 and 51% in 2005. We can observe that spam link generators with $\mathbf{RT} > 0$ appear in both years. The 12% of spam generators of 2004 has $\mathbf{RT} > 0$, and the 27% of generators does in 2005.

We manually checked the hosts that generated the most links in order to understand which type of hosts can be the spam link generators. After the investigation, those hosts in 2004 are turned out to be members of link farms. They have similar hostnames that seem to be generated automatically. On the other hand, the type of such hosts is quite different in 2005. We found that some of them are hosts from university blog communities and homepage hosting companies.

We observed many non-spam hosts were spam link generators in both 2004 and 2005. This implies that normal hosts which cannot be detected by anti-spam techniques can be identified by our approach if it generate links to spam in the future.

4.4 Classification Result

After we obtain spam link generators in 2004 and 2005, we train a classifier using them as positive samples, and test its performance. We also investigate the effectiveness of each feature described in Section 3.

Our spam link generator detection method is based on

Table 6: Performance of spam link generator classification in 2004 and 2005

	2004			2005		
Feature	Precision	Recall	F-measure	Precision	Recall	F-measure
All	**0.725**	0.631	**0.675**	**0.558**	0.534	**0.542**
-W	0.715	0.620	0.663	0.550	0.521	0.532
-S	0.766	0.601	0.673	0.527	0.422	0.461
-RT	0.698	0.632	0.663	0.465	0.460	0.458
-W,-S,-RT	0.755	0.587	0.661	0.576	0.319	0.406
-PR	0.686	0.572	0.624	0.540	0.478	0.504
-oW	0.730	0.627	0.674	0.613	0.448	0.511
-oS	0.670	0.637	0.653	0.629	0.337	0.438
-HJ	0.721	0.630	0.672	0.526	0.536	0.530
-oW,-oS,-HJ	0.668	0.612	0.638	0.333	0.455	0.330
-iW	0.713	0.625	0.666	0.556	0.522	0.535
-iS	0.735	0.608	0.665	0.606	0.394	0.477
-iW, -iS	0.715	0.598	0.651	0.609	0.386	0.473

W : the white score **S** : the spam score **RT** : relative trust
PR : the PageRank score **HJ** : the hijacked score
oW : the total number, the summation and the average of **RT** of white out-neighbors
oS : the total number, the summation and the average of **RT** of spam out-neighbors
iW :the total number, the summation and the average of **RT** of white in-neighbors
iS :the total number, the summation and the average of **RT** of spam in-neighbors
$-feature$ implies we train the classifier without that feature

the machine learning approach. With the features that described in Section 3, we build a classifier using PA-I algorithm [17] provided by the online learning library, oll implement [21]. Five-fold cross validation is used for all classifiers. Hosts are partitioned into five subsets with the same size. The classifier is trained with 4 out of 5 subsets and tested using the rest subset. This process is repeated 5 times, and the average of evaluation metrics from each process will be the performance of the classifier. The order of training samples is shuffled during each iteration.

We adjust the iteration times and parameter of our classifier to achieve the best performance. The classifier of 2004 is trained using 30 iterations and aggressiveness parameter 0.001. The classifier of 2005 is trained using 150 iterations and aggressiveness parameter 0.01(See Section 3). Since the number of spam link generators of 2005 is smaller than that of 2004, we make the proportion of negative sample set to the correct set similar to that of 2004.

4.4.1 Overall performance

Classification results are shown in Table 6. The performance of a spam link generator classifier using whole features is evaluated. Precision over 70% and F-measure over 0.65 are achieved in 2004. In 2005, precision and F-measure are lower than those of 2004, but we can still detect spam link generators with precision over 50%. When we consider the number of spam link generators in the Web, this performance is much better than that of the random selection.

4.4.2 Effectiveness of features

To understand which feature is most effective to classify the spam link generator, a feature ablation study is employed. We remove specific features and observe the change in the performance. If the absence of one feature affects the performance of the classifier than others, that feature might be more important for the classification. The effectiveness of self-related features, out-neighbor-related features, and in-neighbor-related features are observed.

As described in Table 6, the F-measure decreases most when we remove PageRank score from training features for the classifier of 2004. The second most effective feature group is out-neighbor-related features. In particular, spam out-neighbors contribute to the classifier most. In-neighbor-related features are the third effective group, followed by self-related features like white and spam scores. From this, it can be said a host with a high PageRank score and many spam out-neighbors is likely to be a spam link generator in 2004.

In 2005, the most effective feature group is spam out-neighbor-related features. Both precision and F-measure decrease by 20% when we remove feature **oW**, **oS** and hijacked score. Note that the PageRank score is less effective than spam score and **RT** compared to the result of 2004. Considering that more spam link generators of 2005 have positive **RT** than those of 2004 and URLs of top spam generators are related with blog and hosting service, it can be said that many non-spam hosts generated spam links by link hijacking in 2005.

After the ablation study, we found that the PageRank score and spam out-neighbor-related features contribute to the performance of classifier. Among the features related to spam out-neighbors, the number of spam out-neighbors of spam link generators differentiates them from non-generators. Figure 4 and Figure 5 show the distributions of these features. Spam link generators seem to have a higher PageRank score and/or many spam out-neighbors. This agrees with the result obtained by our classifier.

5. CONCLUSION AND FUTURE WORK

Spam detection is a challenging task because existing anti-spam techniques rely on only already-known spamming techniques. In this paper, we focused on spam link generators

that generate links to spam hosts during a specific time interval. By monitoring them, emerging spams can be found promptly. Using various link-based features and machine learning algorithms, we built a classifier to extract spam link generators from the web. In addition to original PageRank score, modified PageRank based scores such as core-based PageRank are selected as feature. Online learning algorithm is used to deal with large scale of web data and allow an easier update. To evaluate our approach, the experiment was performed on the web archive collected during three years. Results showed that we can identify spam link generators with precision of from 56% to 73%, and F-measure of from 0.54 to 0.68. Moreover, we found that almost all new spam links are created by spam link generators.

For future work, we try to update spam filters using spam link generators in order to detect newer spamming techniques. As the first step for this, we examined various link-based features of spam link generators. We plan to introduce textual features in our future work. It is also an interesting question whether we can find the textual characteristics of them. A careful feature selection, however, will be required, because hosts with widely diverse contents could become spam link generators.

6. REFERENCES

[1] S. Nakamura, S. Konishi, A.Jatowt, H. Ohshima, H. Kondo, T. Tezuka, S. Oyama and K. Tanaka. Trustworthiness Analysis of Web Search Results. In *Proceedings of the11th European Conference on Research and Advanced Technology for Digital Libraries*, 2007.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th international conference on World Wide Web*, 1998.

[3] Web spam challenge, `http://webspam.lip6.fr/wiki/pmwiki.php?n=Main.HomePage`

[4] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of the 26th annual International Conference on Machine Learning*, 2009.

[5] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the 13th international conference on Very large data bases*, 2004.

[6] H. Saito, M. Toyoda, M. Kitsuregawa, and K. Aihara. A large-scale study of link spam detection by graph algorithms. In *Proceedings of the 3rd international workshop on Adversarial Information Retrieval on the Web*, 2007.

[7] Y. Chung, M. Toyoda and M. Kitsuregawa. Detecting link hijacking by web Spammers. In *Proceedings of the 13th Pacific-Asia conference on knowledge discovery and data mining*, 2009.

[8] G. Shen, B. Gao, T.-Y. Liu, G. Feng, S. Song, and H. Li. Detecting link spam using temporal information. In *Proceedings of the 6th International Conference on Data Mining*, 2006.

[9] N. Dai, B. D. Davison, and X. Qi. Looking into the past to better classify web spam. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, 2009.

[10] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *Proceedings of the 7th international workshop on the Web and Databases*, 2004.

[11] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the 1th International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[12] Z. Gyöngyi and H. Garcia-Molina. Link Spam Alliance In *Proceedings of the 31st international conference on Very large Data Bases*, 2005.

[13] V. Krishnan, R. Raj. Web Spam Detection with Anti-TrustRank. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web*, 2006.

[14] A. A. Benczúr, K Csalogány, T Sarlós and M. Uher. SpamRank-fully automatic link spam detection. In *Proceedings of the 1st international workshop on Adversarial information retrieval on the Web*, 2005.

[15] L. Becchetti, C. Castillo, D. Donato, S. Leonardi and R. Baeza-Yates. Link-based characterization and detection of Web spam. In *Proceedings of the 2nd international workshop on Adversarial information retrieval on the Web*, 2006.

[16] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your Neighbors: Web Spam Detection using the Web Topology. In *Proceedings of the of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[17] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz and Y. Singer. Online Passive-Aggressive Algorithms, *Journal of Machine Learning Research, Volume 7.*, pp. 551-585, MIT Press, 2006.

[18] Z. Gyöngyi, P. Berkhin, H. Garcia-Molina and J. Pedersen. Link Spam Detection Based on Mass Estimation. In *Proceedings of the 32nd international conference on Very Large Data Bases*, 2006.

[19] A. Carvalho, P. Chirita, E. Moura and P. Calado. Site level noise removal for search engines. In *Proceedings of the 15th international conference on World Wide Web*, 2006.

[20] X. Qi, L. Nie and B. D. Davison. Measuring similarity to detect qualified links, In *Proceedings of the 3rd international workshop on Adversarial Information Retrieval on the Web*, 2007.

[21] D.Okanohara and K. Ohta. Online Learning Library, `http://code.google.com/p/oll/`

[22] Y. Chung, M. Toyoda and M. Kitsuregawa. A study of link farm distribution and evolution using a time series of Web snapshots. In *Proceedings of the 5th international workshop on Adversarial information retrieval on the Web* , 2009.

[23] M. Najork and J. L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th international conference on World Wide Web*, 2001.