A Survey on Link Based Algorithms for Web Spam Detection

M. Sree Vani\(^a\)^*, R.Bhramaramba\(^b\), D.Vasumati\(^c\) and O. Yaswanth Babu\(^d\)

\(^a\)Dept of CSE, MGIT, Gandipet, Hyderabad -500075  
\(^b\)Dept of IT, GITAM University, Vizag-530045  
\(^c\)Dept of CSE, JNTUH, Hyderabad-500032  
\(^d\)IT Manager, TCS, Gachibowli, Hyderabad-500038

Accepted 24 May 2013, Available online 1 June 2013, Vol.3, No.2 (June 2013)

Abstract

Web spamming techniques aim to achieve undeserved rankings in search results. Existing spam pages cause distrust to search engine results. This is not only wastes the time of visitors, but also wastes lots of search engine resources. Research has been widely conducted on identifying such spam and neutralizing its influence. Spammers use three kinds of spamming techniques to get higher score in ranking. These techniques are Link based techniques, hiding techniques and Content-based techniques. In turn, we perform a sub categorization of link-based category into five groups. These are labels propagation, link pruning, reweighting, labels refinement and graph regularization, and feature based. Experimental results show that some of these techniques are working well and can find spam pages more accurate than the others. This paper performs a survey on Link based algorithms for web spam detection.

Keywords: Web spam Detection, content spam, link spam, cloaking, Hiding Techniques, Manipulating Search Engine, Redirection.

1. Introduction

Internet has become an indispensable method to communicate with each other, because of its popularization, low cost, and fast delivery of message. Internet is also widely used for search engines. Search engines make all the information in hand in few seconds. As the popularization of the search engines the problem also come in light which is Web spam. According to definition (J. Abernethy et al, 2010) - (L. Becchetti et al 2006)), the spammers try to add lots of links to target pages in order to manipulate search engine ranking algorithm. Sometimes spammers create a network of pages that is densely connected to each other, which is called link farms.

(1) Link based techniques, also known as link spam

\(^*\)Corresponding author: M. Sree Vani

Figure 1. Classification of web spam techniques

(2) Hiding technique (L. Becchetti et al 2006) – (A. A. Bencz\'ur et al 2005)): this technique is divided to cloaking and redirection. Starting from the cloaking, spammers try to deliver two different contents to web crawler and normal visitors. A web crawler is a computer program that browses the World Wide Web in a methodical, automated manner. A web crawler has a database, which stores the record of requested pages. Hence, search engine ranking algorithms use web crawlers to evaluate web pages, so spammers try to deliver deceptive version of a URL to
web crawlers to get higher score. Redirection means the web page automatically redirects the browser to another URL as soon as the page is loaded. In this way, the search engine indexes the page, but the user does not notice it.

3) Content-based technique —(A. A. Bencz‘ur et al 2008). T. H. Haveliwala. et al (2002)): the spammers try to retouch the content of target pages. For example a page may consist of all the words in a dictionary, so for an uncommon query this page will be chosen with high score, or will be the only result. Another trick is repeating popular words many times in the target pages in order to increase page’s score in search engine ranking.

This survey has two goals: first, it aims to draw a clear roadmap of algorithms, principles and ideas used for web spam detection. second, it aims to build awareness and stimulate further research in the area of adversarial information retrieval. The rest of this paper is organized as follows. Section 2 discusses preliminaries for web spam detection. Section 3 discusses link based spam detection algorithms. Section 4 compares the existing methods briefly. Finally, the conclusion is given in section 5.

2. Preliminaries for Web Spam Detection

This section discusses preliminaries for web spam detection methods. As stated before, the web spam techniques used by spammers are classified into three big categories; which are link based, hiding and content-based techniques. Based on these techniques, there are different web spam detection methods ((J. Abernethy et al, 2008) - (A. Berman 1987)). We will discuss these detection methods in details in the following subsections.

1) Web Graph Model. We model the Web as a graph G = (V, E) with vertices V, representing web pages and directed weighted edges E, representing hyperlinks between pages. If a web page p has multiple hyperlinks to a page p1, we will collapse all these links into one edge (i, j) ∈ E. Self-loops aren’t allowed. We denote a set of pages linked by a page p as Out (pi) and a set of pages pointing to pi as In (pi). Finally, each edge (i, j) ∈ E can have an associated non-negative weight wi,j. A common strategy to assign weights is wij = 1/|Out (pi)|, though other strategies are possible. For instance, in (J. Abernethy et al, 2008) they assign weights proportional to a number of links between pages. In a matrix notation a web graph model is represented by a transition matrix M defined as

M = {wi,j if (i, j) ∈ e
0 otherwise

2) PageRank (L. Page et al 1998) uses link information to compute global importance scores for all pages on the web. The key underlying idea is that a link from a page pi to a page p1 shows an endorsement or trust of page pi in page p1, and the algorithm follows the repeated improvement principle, i.e. the true score is computed as a convergence point of an iterative updating process. The most popular and simple way to introduce PageRank is a linear system formulation. In this case PageRank vector for all pages on the web is defined as the solution of the matrix equation

\[ \overrightarrow{\pi} = (1 - c) \cdot M \overrightarrow{\pi} + c \cdot \overrightarrow{r} \]  (1)

Where, c is a damping factor, and \( \overrightarrow{r} \) a static PageRank vector. For non-personalized PageRank it is a unit vector (1/N, ..., 1/N), where N = |V|. It is worth noting that this is needed to meet the conditions of a Perron-robenius theorem (A. Berman 1987) and guarantee the existence of a stationary distribution for the corresponding akov chain. It has also a semantic meaning that at any given moment “random surfer” can visit any page with the non-zero probability. For a non-uniform static vector \( \overrightarrow{r} \) the solution is called a personalized PageRank (PPR) ((D. Fogaras et al 2004) - (T. H. Haveliwala. et al 2002)) and the vector \( \overrightarrow{r} \) is called a personalization, random jump or teleportation vector. There are a few useful properties of PageRank. First, it is linear in \( \overrightarrow{r} \), i.e. if \( \overrightarrow{r}_1 \) is the solution of (1) with the personalization vector \( \overrightarrow{r}_2 \) and \( \overrightarrow{r}_3 \) is the solution of (1) with the personalization vector \( \overrightarrow{r}_4 \), then the vector \( \overrightarrow{r}_1 + \overrightarrow{r}_2 \) is the solution for the equation (1) with the personalization vector \( \overrightarrow{r}_3 + \overrightarrow{r}_4 \). As a consequence of that we can expand a personalized PageRank in the following way as in equation (2)

\[ \text{PPR}(\overrightarrow{r}) = \frac{1}{N} \sum_{v \in V} \text{PPR}(X_v) \]  (2)

Where \( X_v \) is the teleportation vector consisting of all zeros except for a node v such that \( X_v(v) = 1 \) (we use this property in Section 3.2.2). Second, PageRank has an interpretation as a probability of a random walk terminating at a given vertex where the length follows the geometric distribution ((D. Fogaras et al (2003) – (G. Jeh et al 2005)), i.e. the probability to make j steps before termination is equal to \( c \cdot (1 - c)^j \) and the following representation is valid

\[ \text{PPR}(\overrightarrow{r}) = c \cdot (1 - c)^j \sum_{v \in V}(1 - c)^j (M^T) \]  (3)

3) HITS (J. M. Kleinberg et al (1999)) algorithm assigns hub and authority scores for each page on the web and is based on the following observation. Page pi is a good hub, has a high hub score hi, if it points to many good (authoritative) pages; and page is a good authority if it is referenced by many good hubs, and therefore has a high authority score ai. We see the algorithm also has a repeated improvement principle behind it. In its original form the algorithm considers pages relevant to a query based on a keyword-based ranking (the root set), all the pages that point to them, and all the pages referenced by these pages. For this subset of the Web an adjacency matrix is defined, denoted as A. The corresponding authority and hub scores for all pages from the subset are formalized in the following pair of equations:

\[ \overrightarrow{a} = A^T \overrightarrow{h}, \]
\[ \overrightarrow{h} = A \overrightarrow{a} \]  (4)

It can be shown that the solution \( (\overrightarrow{h}, \overrightarrow{a}) \) for the system of equations (4) after repetitive updating converges to the principal eigenvectors of AAT and ATA correspondingly. (A. Y. Ng et al 2001) Studies the robustness of PageRank and HITS algorithms with respect to small perturbations. Specifically, they analyzed how severely the ranks will
change if a small portion of the Web is modified (removed). They report that PageRank is sable to small perturbations of a graph, while HITS is quite sensitive. (M. Bianchini et al 2008) Conducts a comprehensive analysis of PageRank properties and how link farms can affect the ranking. They prove that for any link farm and any set of target pages the sum of PageRank scores over the target pages is at least a linear function of the number of pages in a link farm. The optimality properties of link farms have been studied by (Z. Gy"ongyi et al 2005). They derive the boosting factor for one-target spam farm and prove that this farm is optimal if all the boosting pages in a farm have no links between them, they point to the target page and target page links back to a subset of them. They also discuss motivations of spammers to collude (form alliances), and study optimality of web spam rings and spam quasi cliques. There is also a relevant work which analyzes properties of personalized PageRank (A. Langville et al 2004) and a survey on efficient PageRank computation (P. Berkhin et al 2005).

3. Link-based Algorithms for web spam detection

3.1. Algorithms based on labels propagation

The key idea behind algorithms from this group is to consider a subset of pages on the web with known labels and then compute labels of other nodes by various propagation rules. One of the first algorithms from this category is TrustRank (Z. Gy"ongyi et al 2004), which propagates trust from a small seed set of good pages via a personalized PageRank. The algorithms rely on the principle of approximate isolation of a good set –good pages point mostly to good pages. To select a seed set of reputable pages they suggest using an inverse PageRank, which operates on a graph with all edges reversed. Having computed inverse PageRank score for all pages on the Web, they take top-K pages and let human annotator to judge on reputation of these pages. Then they construct a personalization vector where components corresponding only to reputable judged pages are non-zero. Finally, personalized PageRank is computed. TrustRank shows better properties than PageRank for web spam demotion. The follow up work on trust propagation is Anti-TrustRank (V. Krishnan et al 2006). Opposite to TrustRank they consider distrust propagation from a set of known spam pages on an inverted graph. A seed set is selected among pages with high PageRank values. They found that their approach outperforms TrustRank on the task of finding spam pages with high precision and is able to capture spam pages with higher PageRank values than TrustRank. There is also an algorithm, called Bad-Rank M. Sobek et al (2002), which proposes the idea to compute badness of a page using inverse PageRank computation. One can say that the relation between PageRank and TrustRank is the same as between BadRank and Anti-TrustRank. (B. Wu et al 2006) further researches the idea of propagation by analyzing how trust and distrust propagation strategies can work together. First, they challenge the way trust is propagated in TrustRank algorithm – each child18 gets an equal part of parent’s trust (c · TR(p)) and propose two more strategies:

- constant splitting, when each child gets the same discounted part of parent’s trust c · TR(p) score without respect to number of children;
- logarithmic splitting, when each child gets an equal part of parent’s score normalized by logarithm of number of children (c · log(1+|out(p)|)/). They also analyze various partial trust aggregation strategies, whereas TrustRank simply sums up trust values from each parent. Specifically, they consider maximum share strategy, when the maximum value sent by parents is used; and maximum parent strategy, when propagation is performed to guarantee that a child score wouldn’t exceed maximum of parent’s scores. Finally, they propose to use a linear combination of trust and distrust values:

$$\text{TotalScore}(p) = \eta \cdot \text{TR}(p) - \beta \cdot \text{AntiTR}(p),$$

Where $\beta, \eta \in (0, 1)$. According to their experiments, combination of both propagation strategies result in better spam demotion (80% of spam sites disappear from the top ten buckets in comparison with the TrustRank and PageRank), maximum share with logarithmic splitting is the best way to compute trust and distrust values. The idea of trust and distrust propagation in the context of reputation systems was studied in (R. Guha et al 2004). Two algorithms (A. A. Bencz"ur et al 2005, Z. Gyongyi et al 2006) utilize PageRank decomposition property to estimate the amount of undeserved PageRank coming from suspicious nodes. In (A. A. Bencz"ur et al 2005) the SpamRank algorithm is proposed; it finds supporters for a page using Monte Carlo simulations (D. Fogaras et al 2004), assigns a penalty score for each page by analyzing whether personalized PageRank score PRP(X\i) is distributed with the bias towards suspicious nodes, and finally computes SpamRank for each page as a PPR with the personalization vector initialized with penalty scores. The essence of the algorithm is in the penalty scores assignment. Authors partition all supporters for a page by their PageRank scores using binning with exponentially increasing width, compute the correlation between the index of a bin and the logarithm of a count in the bin, and then assign penalty to supporters by summing up correlation scores of pages which they support. The insight behind the proposed approach is that PageRank follows power law distribution (G. Pandurangan et al 2002). The concept of spam mass was introduced in (Z. Gyongyi et al 2006). Spam mass measures the amount of PageRank that comes from spam pages. Similar to TrustRank it needs the core of known good pages to estimate the amount of PageRank coming from good pages. The algorithm works in two stages. First, it computes PageRank $\pi$ and TrustRank $\pi'$ vectors and estimates the amount of spam mass for each page using the formula

$$m = \frac{n - n'}{n}$$

(6)
Second, the threshold decision, which depends on the value of spam mass, is made. It is worth noting that the algorithm can effectively utilize knowledge about bad pages. In this paragraph we will refer to out-neighbors of a page as children and in-neighbors as parents. Credibility-based link analysis is described in (J. Caverlee et al. 2007). In this work the authors define the concept of k-Scoped Credibility for each page, propose several methods of its estimation, and show how it can be used for web spam detection. Specifically, they first define the concept of Bad Path, a k-hop random walk starting from a current page and ending at a spam page, and then compute the tuned k-Scoped Credibility score as

$$C_k(p) = \left\{ 1 - \sum_{i=1}^{k} \sum_{\text{path} \in \text{BadPath}(p)} P(\text{path}_i(p)) \right\} \cdot y(\text{path}(p))$$  \hspace{1cm} (7)

where \( k \) is a parameter specifying the length of a random walk, \( y(\text{path}(p)) \) is a credibility penalty factor that is needed to deal with only partial knowledge of all spam pages on the Web, and \( P(\text{path}_i(p)) = \prod_{l=1}^{i} w_{li+1} \). \hspace{1cm} (8)

The credibility score can be used to down weight or prune low credible links before link-based ranking or to change the personalization vector in PPR, TrustRank, or Anti-TrustRank. In (A. Joshi et al. 2007) the concept of anchor is defined, as a subset of pages with known labels, and various anchor-based proximity measures on graphs are studied. They discuss personalized PageRank; harmonic rank, which is defined via random walk on a modified graph with an added source and a sink such that all anchor vertices are connected to a source and all vertices are connected to a sink with probability \( c \), nonconserving rank, which is a generalization of personalized PageRank satisfying the equation

$$\vec{\pi} = (1 - (1 - c) \cdot \vec{M}^T)^{-1} \vec{\tau}$$ \hspace{1cm} (9)

They report that non-conserving rank is the best for trust propagation, while harmonic rank better suits for distrust propagation. Spam detection algorithm utilizing pages similarity is proposed in (A. Benczúr, et al. 2006), where similarity-based top-K lists are used to compute a spam score for a new page. Authors consider co-citation, CompanionRank, SimRank (G. Jeh et al. 2002), and kNN-SVD projections as methods to compute similarity between pages. First, for a page to be classified a top-K result list is retrieved using some similarity measure. Second, using the retrieved pages the following four values are computed: fraction of the number of labeled spam pages in the list (SR), a number of labeled spam pages divided by a number of labeled good pages in the list (SON), sum of the similarity values of labeled spam pages divided by the total similarity value of pages retrieved (SVR), and the sum of the similarity values of labeled spam pages divided by the sum of the similarity values of labeled good pages (SVONV). Third, threshold-based rule is used to make a decision. According to their experiments, similarity-based spam detection (SVR, SVONV) performs better at high levels of recall, while Anti-TrustRank (V. Krishnan et al. 2006) and combined Trust-Distrust (B. Wu et al. 2006) algorithms show higher precision at low recall levels. The seminal line of work was done by Baeza-Yates et al. (R. Baeza-Yates et al. 2006, L. Becchetti et al. 2006, L. Becchetti et al. 2008, L. Becchetti et al. 2006). In (R. Baeza-Yates et al. 2006), inspired by the PageRank representation (Equation 8), they propose the concept of functional rank, which is a generalization of PageRank via various damping functions. They consider ranking based on a general formula

$$\vec{p} = \frac{1}{n} \vec{\tau} \sum_{j=0}^{\infty} \text{damping}(j) (\vec{M}^T)^j$$ \hspace{1cm} (10)

and prove the theorem that any damping function such that the sum of damping’s is 1 yields a well-defined normalized functional ranking. They study exponential (PageRank), linear, quadratic hyperbolic (TotalRank), general hyperbolic (HyperRank) damping functions, and propose efficient methods of rank computation. In (L. Becchetti et al. 2006) they research the application of general damping functions for web spam detection and propose truncated PageRank algorithm, which uses truncated exponential model. The key underlying observation behind the algorithm is that spam pages have a large number of distinct supporters at short distances, while this number is lower than expected at higher distances. Therefore, they suggest using damping function that ignore the direct contribution of the first few levels of in-links

$$\text{damping}(j) = \begin{cases} 0 & \text{if } j \leq 1 \\ \frac{n}{n-1} & \text{otherwise} \end{cases}$$ \hspace{1cm} (11)

In this work they also propose a probabilistic counting algorithm to efficiently estimate number of supporters for a page. Link-based feature analysis and classification models using link-based and content-based features are studied in (L. Becchetti et al. 2006, L. Becchetti et al. 2008) correspondingly.

### 3.2 Link pruning and reweighting algorithms

Algorithms belonging to this category tend to find unreliable links and demote them. The seminal work (K. Bharat et al. 1998) raises problems in HITS (J. M. Kleinberg et al. 1999)) algorithm, such as domination of mutually reinforcing relationships and neighbor graph topic drift, and proposed methods of their solution by augmenting a link analysis with a content analysis. They propose to assign each edge an authority weight of \( 1/k \) if there are \( k \) pages from one site link to a single page on another site, and assign a hub weight of \( 1/l \) if a single page from the first site is pointing to \( l \) pages on the other site. To combat against topic drift they suggest using query expansion, by taking top-\( K \) frequent words from each initially retrieved page and candidate page set pruning, by taking page relevance as a factor in HITS computation. The same problems are studied in (S. Nomura et al. 2004), where a projection-based method is proposed to compute authority scores. They modify eigenvector part of HITS algorithm in the following way. Instead of computing a principal eigenvector of ATA, they compute all eigenvectors of the matrix and then take the eigenvector with the biggest projection on the root set (set of pages
originally retrieved by keyword search engine, as in HITS; finally they report authority scores as the corresponding components of this eigenvector.

Another group introduces the concept of tightly-knit community (TKC) and proposes SALSA algorithm (R. Lempel et al 2001), which performs two random walks to estimate authority and hub scores for pages in a sub graph initially retrieved by keyword based search. It is worth noting that the original and an inverted sub graphs are considered to get two different scores. An extension of this work (G. Roberts et al 2003) considers clustering structure on pages and their linkage patterns to down weight bad links. The key trick is to count number of clusters pointing to a page instead of number of individual nodes. In this case authority of a page is defined as follows:

\[ a_i = \sum_{j \in t(k)} \frac{1}{\sum_{j \in t(i)} s_{jk}} \]  \( (12) \)

\[ s_{ik} = \frac{1}{\sum_{j \in t(i)} s_{jk}} \]  And \( l(i) \) is a set of pages linked from page \( p_i \).

The approach acts like popularity ranking methods discussed in (A. Borodin et al 2011, S. Chakrabarti et al 2002)). (L. Li et al 2002) studies “small-in-large-out” problem of HITS and proposes to reweight incoming and outgoing links for pages from the root set in the following way. If there is a page whose in-degree is among the three smallest ones and whose out degree is among the three largest ones, then set the weight 4 for all in-links of all root pages, otherwise set to 1. Run one iteration of HITS algorithm without normalization. Then if there exists a root page whose authority value is among the three smallest ones and whose hub value is among the three largest ones, set the weight 4 for all in-links of all root pages, and then run the HITS algorithm again. B. Davison et al (2000) introduces the concept of “neponistic” links – links that present for reasons other than merit, for instance, navigational links on a website or links between pages in a link farm. They apply C4.5 algorithm to recognize neponistic links using 75 different binary features such as IsSimilarHeaders, IsSimilarHost, is number of shared in-links is greater than a threshold. Then they suggest pruning or down weighting neponistic links. In their other work (B. Wu et al 2005) they continue studying links in densely connected link farms. The algorithm operates in three stages. First, it selects a set of bad seed pages guiding by the definition that a page is bad if intersection of its incoming and outgoing neighbors is greater than a threshold. Second, it expands the set of bad pages following the idea that a page is bad if it points to lot of bad pages from the seed set. Finally, links between expanded set of bad pages are removed or down weighted and any link-based ranking algorithm (L. Page et al 1998, G. Karypis et al 1998) can be applied. Similar ideas are studied on a host level in A. L. da Costa Carvalho et al 2006). In (B. Wu et al 2006) the concept of a complete hyperlink is proposed, a hyperlink coupled with the associated anchor text, which is used to identify pages with suspiciously similar linkage patterns. Rationale behind their approach is that pages that have high complete hyperlink overlap are more likely to be machine-generated pages from a link farm or pages with duplicating content. The algorithm works as follows. First, it builds a base set of documents, as in HITS, and generates a page-hyperlink matrix using complete hyperlinks, where \( \lambda_0 = 1 \), if a page \( p_i \) contains complete-hyperlink \( c_i \). Then it finds bipartite components with the size greater than a threshold in the corresponding graph, where parts are pages and links, and down weight complete hyperlinks from large components. Finally, a HITS-like algorithm is applied on a reweighted adjacency matrix. (H. Zhang et al 2004) notices that PageRank score of pages that achieved high ranks by link-spamming techniques correlates with the damping factor \( c \). Using this observation authors identify suspicious nodes, whose correlation is higher than a threshold and down weight outgoing links for them with some function proportional to correlation. They also prove that spammers can amplify PageRank score by at most 1c and experimentally show that even two-node collusion can yield a big PageRank amplification. The follow-up work (R. Baeza-Yates et al 2005) performs more general analysis of different collusion topologies, where they show that due to the power law distribution of PageRank (G. Pandurangan et al 2002), the increase in PageRank is negligible for top-ranked pages. The work is similar to (Z. Gyongyi et al 2005, S. Adali et al 2005).

3.3 Algorithms with link-based features

Algorithms from this category represent pages as feature vectors and perform standard classification or clustering analysis. (E. Amitay et al 2003) studies link-based features to perform website categorization based on their functionality. Their assumption is that a site sharing similar structural patterns, such as average page level or number of out links per leaf page, share similar roles on the Web. For example, web directories mostly consist of pages with high ratio of outlinks to inlinks, form a tree-like structure, and the number of outlinks increases with the depth of a page; while spam sites have specific topologies aimed to optimize PageRank boost and demonstrate high content duplication. Overall, each website is represented as a vector of 16 connectivity features and a clustering is performed using cosine as a similarity measure. Authors report that they managed to identify 183 web spam rings forming 31 cluster in a dataset of 1100 sites. Numerous link-based features, derived using PageRank, TrustRank, and truncated PageRank computation are studied in (L. Becchetti et al 2006). Mixture of content-based and link-based features is used to combat against web spam in (I. Drost et al 2005, L. Becchetti et al 2008), spam in blogs (P. Kolari et al 2006, Y.-R. Lin et al 2007).

3.4 Algorithms based on labels refinement

The idea of labels refinement has been studied in machine learning literature for general classification problems for a long time. In this section we present algorithms that apply this idea for web spam detection. In (C. Castillo et al 2007) a few web graph-based refinement strategies are proposed. The algorithm in (C. Castillo et al 2007) works
in two stages. First, labels are assigned using a spam detection algorithm discussed in (L. Becchetti et al 2006). Then, at the second stage labels are refined in one of three ways. One strategy is to perform Web graph clustering (G. Karypis et al 1998) and then refine labels guided by the following rules. If the majority of pages in a cluster is predicted to be spam, they denote all pages in the cluster as spam. Formally, they assume that predictions of the base algorithm are in [0, 1], then they compute the average value over the cluster and compare it against a threshold. The same procedure is done for non-spam prediction. The other strategy of label refinement, which is based on propagation with random walks, is proposed in (D. Zhou et al 2010). The key part is to initialize the personalization vector \( \tilde{r} \) in PPR by normalizing the predictions of the base algorithm:

\[
\tilde{r}_p = \frac{s(p)}{\sum_{p \in S} s(p)}
\]

(13)

Where \( s(p) \) is a prediction of the base algorithm and \( \tilde{r}_p \) is the component of the vector \( \tilde{r} \) corresponding to page \( p \). Finally, the third strategy is to use stacked graphical learning (Z. Kou et al 2007). The idea is to extend the original feature representation of an object with a new feature which is an average prediction for related pages in the graph and run a machine learning algorithm again. They report 3\% increase over the baseline after two rounds of stacked learning. A few other re-labeling strategies are proposed in (Q. Gan et al 2007, G. Geng et al 2008, and G. Geng et al 2009). (Q. Gan et al 2007) Suggests constructing an absolutely new feature space by utilizing predictions from the first stage: the label by the base classifier, the percentage of incoming links coming from spam and percentage outgoing links pointing to spam, etc. Overall, they consider seven new features and report increase in performance over the base classifier. (G. Geng et al 2008) proposes to use feature re-extraction strategy using clustering, propagation and neighbor-graph analysis. Self-training was applied in (G. Geng et al 2009) so as to reduce the size of the training dataset requiring for web spam detection.

3.5 Graph regularization algorithms

Algorithms in this group perform transductive inference and utilize Web graph to smooth predicted labels. According to experimental results, graph regularization algorithms for web spam detection can be considered as the state-of-the-art at the time of writing. The work (D. Zhou et al 2011) builds a discrete analogue of classification regularization theory (A. Tikhonov et al 1977, G. Wahba et al. 1990) by defining discrete operators of gradient, divergence and Laplacian on directed graphs and proposes the following algorithm. First, they compute an inverse weighted PageRank with transition probabilities defined as \( a_{ij} = \frac{w_{ij}}{\ln(p_i)} \). Second, they build the graph Laplacian

\[
L = \Pi - \alpha \frac{\Pi A^T \Pi}{2}
\]

(14)

where \( \alpha \) is a user-specified parameter in [0, 1], \( A \) is a transition matrix, and \( \Pi \) is a diagonal matrix with the PageRank score 20 over the diagonal. Then, they solve the following matrix equation

\[
\hat{y} = \Pi \hat{y}
\]

(15)

where \( \hat{y} \) is a vector consisting of \([-1, 0, 1]\) such that \( y_i = 1 \) if page is normal, \( y_i = -1 \) if it is spam, and \( y_i = 0 \) if the label for a page is unknown. Finally, the classification decision is made based on the sign of the corresponding component of vector \( \hat{y} \). It is worth noting that the algorithm requires strongly connected graphs. Another algorithm that follows regularization theory is described in (J. Abernethy et al, 2010). There are two principles behind it. First, it addresses the fact that hyperlinks are not placed at random implies some degree of similarity between the linking pages (B. D. Davison et al. 2000, S. Chakrabarti et al (2002)) This, in turn, motivates to add a regularizer to the objective function to smooth predictions. Second, it uses the principle of approximate isolation of good pages that argues for asymmetric regularizer. The final objective function has the following form:

\[
\Omega(\hat{w}_i, \hat{z}_i) = \frac{1}{2} \sum_{i=1}^n L_i \hat{w}_i + \lambda_1 \| \hat{w}_i \|^2 + \lambda_2 \| \hat{z}_i \|^2 + \gamma \sum_{i,(j)E} a_{ij} \Phi(\hat{w}_j \hat{z}_i + z_i, \hat{w}_j \hat{z}_i + \hat{z}_i)
\]

(16)

where \( L(a, b) \) is a standard loss function, \( \hat{w}_i \) is a vector of coefficients, \( \hat{z}_i \) and \( y_i \) are feature representation and a true label correspondingly, \( z_i \) is a bias term, \( a_{ij} \) is a weight of the link \((i, j) \in E\), and \( \Phi(a, b) = \max(0, b - a) \| \hat{w}_j \hat{z}_i + z_i, \hat{w}_j \hat{z}_i + \hat{z}_i \) is the regularization function. The authors provide two methods to find the solution of the optimization problem using conjugate gradient and alternating optimization. They also study the issue of weights setting for a host graph and report that the logarithm of the number of links yields the best results. Finally, according to the experimental study, the algorithm has good scalability properties. Some interesting idea to extract web spam URLs from SEO forums is proposed in (Z. Cheng et al 2011). The key observation is that on SEO forums spammers share links to their websites to find partners for building global link farms. Researchers propose to solve a URL classification problem using features extracted from SEO forum, from Web graph, and from a website, and regularizing it with four terms derived from link graph and user-URL graph. The first term is defined as follows. Authors categorize actions on a SEO forum in three groups: post URL in a root of a thread (weight 3), URL in reply (weight 2), view URL in previous posts (weight 1). Then they build a user-URL bipartite graph where an edge weight is the sum of all weights associated with the corresponding actions. After that they compute SimRank for all pairs of URLs and introduce the regularization term via Laplacian of the similarity matrix. The second regularization term is defined analogously, but now simply via a Laplacian of a standard transition matrix. Third and fourth asymmetric terms, defined via the Web graph transition matrix and its diagonal row or column aggregated matrices, are introduced to take into account the principle of approximate isolation of good pages. Finally, the sound quadratic problem is solved. Authors report that even legitimately looked spam websites can be effectively
detected using this method and hence the approach complements the existing methods.

4. Comparison of Existing Methods

Each of the above explained methods will add special key words or terms to web pages to fool search engine’s ranking algorithms. In link-based techniques, spammers try to fool by adding many links to special pages or placing links from good pages. In hiding techniques, spammers deliver different contents to web crawler and normal visitors, so there is a difficulty to recognize this kind of spamming technique. The last method is using the features of web pages to fool search engines. Between these methods, hiding techniques are harder to defend since spammers deliver different contents to web crawlers and web browsers, detecting this kind of cloaking needs comparison between the content of the page sent to web crawler and normal visitor. This comparison is time consuming. Also there is no automatic method in this area to find cloaking or redirection. Content-based techniques have the most destructive effect on search engines. Since spammers can easily add key words or special terms visibly or invisibly to web pages, this method is a popular one between spammers. In addition, if we search a query in a popular search engine like Google, the first spamming technique that we encounter will be content-based techniques. According to experimental results achieved from different proposed methods in different areas, methods proposed in hiding techniques can find spam pages better than other methods. These methods can find more spam pages with high precision and reasonable recall. Consequently, these methods will help search engines more.

5. Conclusion

As a solution for web spam issue, web spam detection methods have been proposed. These methods will help search engine ranking algorithms allocate ranking score more accurate than before. Therefore, spam pages will get lower score. In this paper we surveyed existing techniques and algorithms created to fight against web spam. Furthermore, current anti-spam algorithms show a competitive performance in detection, about 90%, that demonstrates the successful results of many researchers. However, we cannot stop here because spam is constantly evolving and still negatively affects many people and businesses. We believe that there is still high need for researching in this area to help search engine’s ranking algorithms for developing efficient and effective methods to find spam pages in the future.

References


S. Chakrabarti(2002) Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann


