Survey on Web Spam Detection: Principles and Algorithms

Nikita Spirin
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
spirin2@illinois.edu

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
hanj@cs.uiuc.edu

ABSTRACT

Search engines became a de facto place to start information acquisition on the Web. Though due to web spam phenomenon, search results are not always as good as desired. Moreover, spam evolves that makes the problem of providing high quality search even more challenging. Over the last decade research on adversarial information retrieval has gained a lot of interest both from academia and industry. In this paper we present a systematic review of web spam detection techniques with the focus on algorithms and underlying principles. We categorize all existing algorithms into three categories based on the type of information they use: content-based methods, link-based methods, and methods based on non-traditional data such as user behaviour, clicks, HTTP sessions. In turn, we perform a subcategorization of link-based category into five groups based on ideas and principles used: labels propagation, link pruning and reweighting, labels refinement, graph regularization, and feature-based. We also define the concept of web spam numerically and provide a brief survey on various spam forms. Finally, we summarize the observations and underlying principles applied for web spam detection.

Keywords

web spam detection, content spam, link spam, cloaking, collusion, link farm, pagerank, random walk, classification, clustering, web search, user behaviour, graph regularization, labels propagation

1. INTRODUCTION

Spam pervades any information system, be it e-mail or web, social, blog or reviews platform. The concept of web spam or *spamdexing* was first introduced in 1996 [31] and soon was recognized as one of the key challenges for search engine industry [57]. Recently [50; 97], all major search engine companies have identified adversarial information retrieval [41] as a top priority because of multiple negative effects caused by spam and appearance of new challenges in this area of research. First, spam deteriorates the quality of search results and deprives legitimate websites of revenue that they might earn in the absence of spam. Second, it weakens trust of a user in a search engine provider which is especially tangible issue due to zero cost of switching from one search provider to another. Third, spam websites serve as means of malware and adult content dissemination and fishing attacks. For instance, [39] ranked 100 million pages using PageRank algorithm [86] and found that 11 out of top 20 results were pornographic websites, that achieved high ranking due to content and web link manipulation. Last, it forces a search engine company to waste a significant amount of computational and storage resources. In 2005 the total worldwide financial losses caused by spam were estimated at $50 billion [63], in 2009 the same value was estimated already at $130 billion [64]. Among new challenges which are emerging constantly one can highlight a rapid growth of the Web and its heterogeneity, simplification of content creation tools (e.g., free web wikis, blogging platforms, etc.) and decrease in website maintenance cost (e.g., domain registration, hosting, etc.), evolution of spam itself and hence appearance of new web spam strains that cannot be captured by previously successful methods.

Web spam phenomenon mainly takes place due to the following fact. The fraction of web page referrals that come from search engines is significant and, moreover, users tend to examine only top ranked results. Thus, [98] showed that for 85% of the queries only the first result page is requested and only the first three to five links are clicked [65]. Therefore, inclusion in the first SERP[1] has a clear economic incentive due to an increase in website traffic. To achieve this goal website owners attempt to manipulate search engine rankings. This manipulation can take various forms such as the addition of a surrogate content on a page, excessive and undeserved link creation, cloaking, click fraud, and tag spam. We define these concepts in Section 2, following the work [54; 82; 13; 57]. Generally speaking, web spam manifests itself as a web content generated deliberately for the purpose of triggering unjustifiably favourable relevance or importance of some web page or pages [54]. It is worth mentioning that the necessity of dealing with the malicious content in a corpus is a key distinctive feature of adversarial information retrieval [41] in comparison with the traditional information retrieval, where algorithms operate on a clean benchmark data set or in an intranet of a corporation.

According to various studies [85; 25; 12] the amount of web spam varies from 6 to 22 percent, which demonstrates the scope of the problem and suggests that solutions that require manual intervention will not scale. Specifically, [7] shows that 6% of English language web pages were classified as spam, [25] reports 22% of spam on a host level and [12] estimates it as 16.5%. Another group of researchers study not only the cumulative amount of spam but its dis-

[1]Search Engine Result Page.

tribution among countries and top level domains [85]. They report 13.8% of spam in the English speaking internet, 9% in Japanese, 22% in German, and 25% in French. They also show that 70% of pages in the *.biz domain and 20% in the *.com domain are spam.

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. To the best of our knowledge there is no comprehensive but concise web spam mining survey with the focus on algorithms yet. This work complements existing surveys [54; 13; 82; 57] and a book [23] on the topic of web spam. [54] presents a web spam taxonomy and provides a broad coverage of various web spam forms and definitions. In [13] issues specific to web archives are considered. [82; 57] enumerate spam forms and discuss challenges caused by spam phenomenon for web search engines. [23] is a recent book, which provides the broadest coverage of web spam detection research available so far. We think that parallel research on email spam fighting [19; 95; 113] and spam on social websites [58] might also be relevant. [33] presents a general framework for adversarial classification and approaches the problem from game-theoretical perspective.

The paper is organized as follows. In Section 2 we provide a brief overview of web spam forms following [54; 82; 13; 57]. Then we turn to content-based mining methods in Section 3.1. In Section 3.2 we provide a careful coverage of link-based spam detection algorithms. In Section 3.3 we consider approaches to fight against spam using clickthrough and user behaviour data, and by performing realtime HTTP sessions analysis. Finally, we summarize key principles underlying web spam mining algorithms in Section 4 and conclude in Section 5.

2. WEB SPAM TAXONOMY

2.1 Content Spam

The content spam is probably the first and most widespread form of web spam. It is so widespread because of the fact that search engines use information retrieval models based on a page content to rank web pages, such as a vector space model [96], BM25 [94], or statistical language models [114]. Thus, spammers analyze the weaknesses of these models and exploit them. For instance, if we consider TFIDF scoring

$$TFIDF(q,p) = \sum_{t \in q \wedge t \in p} TF(t) \cdot IDF(t), \qquad (1)$$

where q is a query, p is a page, and t is a term, then spammers can try to boost TF of terms appearing on a page[2]. There are multiple facets based on which we can categorize content spam.

Taking a document structure into account there are 5 subtypes of content spamming.

- Title Spamming. Due to high importance of the title field for information retrieval [79] spammers have a clear incentive to overstuff it so as to achieve higher overall ranking.
- Body Spamming. In this case the body of a page is modified. This is the most common form of content spam because it is cheap and simultaneously allows applying various strategies. For instance, if a spammer wants to achieve a high ranking of a page for only limited predefined set of queries, they can use the repetition strategy by overstuffing body of a page with terms that appear in the set of queries (there even was a spamming competition to rank highest for the query "nigritude ultramarine" [38]). On the other hand, if the goal is to cover as many queries as possible, the strategy could be to use a lot of random words at once. To hide part of a content used to boost ranking, site owners make it coloured in the same way as a background so that only machines can recognize it.
- Meta-Tags Spamming. Because meta-tags play a specific role in a document description, search engines analyze them carefully. Hence, the placement of spam content in this field might be very prospective from spammer point of view. Because of the heavy spamming, nowadays search engines give a low priority to this field or even ignore it completely.
- Anchor Text Spamming. The usefulness of anchor text[3] for web ranking was first introduced in 1994 [80], and instantly spammers added the corresponding strategy to their "arsenal". Spammers create links with the desired anchor text (often unrelated to a linking page content) in order to get the "right" terms for a target page.
- URL Spamming. Some search engines also consider a tokenized URL of a page as a zone. And hence spammers create a URL for a page from words which should be mentioned in a targeted set of queries. For example, if a spammer wants to be ranked high for the query "cheap acer laptops", they can create a URL "cheap-laptops.com/cheap-laptops/acer-laptops.html"[4].

With the advent of link-based ranking algorithms [86; 68] content spam phenomenon was partially overcome. However, spam is constantly evolving and soon afterwards spammers started constructing link farms [53; 3], groups of highly interconnected spam pages, with the aim to boost ranking of one or a small number of *target* pages in a farm. This form of spam is referred to as link spam.

2.2 Link Spam

There are two major categories of link spam: outgoing link spam and incoming link spam.

2.2.1 *Outgoing link spam*

This is the easiest and cheapest method of link spam because, first, a spammer have a direct access to his pages and therefore can add any items to them, and second, they can easily copy the entire web catalogue such as DMOZ[5] or Yahoo! Directory[6] and therefore quickly create a large set

[2] we assume that IDF of a term cannot be manipulated by a spammer due to an enormous size of the Web

[3] a visible caption for a hyperlink

[4] it is worth noting that the example also demonstrates idea of keyword repetition, because the word "laptops" appears three times in a tokenized URL representation

[5] dmoz.org

[6] http://dir.yahoo.com/

of authoritative links, accumulating relevance score. Outgoing link spamming targets mostly HITS (Section *3.2.1*) algorithm [68] with the aim to get high hub score.

2.2.2 *Incoming link spam*

In this case spammers try to raise a PageRank (Section *3.2.1*) score of a page (often referred to as a *target* page) or simply boost a number of incoming links. One can identify the following strategies depending on an access to pages.

- **Own Pages.** In this case a spammer has a direct control over all the pages and can be very flexible in his strategies. He can create his own link farm[7] and carefully tune its topology to guarantee the desired properties and optimality. One common link farm has a topology depicted on a fig. 1 and is named as a *honeypot* farm. In this case a spammer creates a page which looks absolutely innocent and may be even authoritative (though it is much more expensive), but links to the spammer's target pages. In this case an organically aggregated PageRank (authority) score is propagated further to target pages and allows them to be ranked higher. More aggressive form of a honeypot schema is *hijacking*, when spammers first hack a reputable website and then use it as a part of their link farm. Thus, in 2006 a website for prospective CS students was hijacked and spammed with link of pornographic nature.

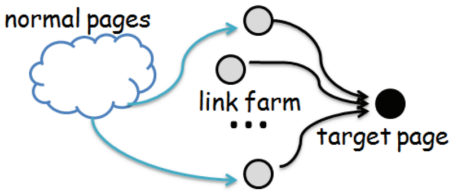

Figure 1: Scheme of a typical link farm.

Spammers can also collude by participating in link exchange schemes in order to achieve higher scale, higher in-link counts, or other goals. Motivations of spammers to collude are carefully analyzed in [53]. Optimal properties of link farms are analyzed in [53; 3; 117].

To reduce time spent on a link farm promotion spammers are also eager to buy expired and abandoned domain names. They are guided by the principle that due to the non-instant update of an index and recrawling, search engines believe that a domain is still under the control of a good website owner and therefore spammers can benefit with "resources" and reputation left by the previous website for some time.

We also consider redirection as an instant type of honeypot scheme[8]. Here the spamming scheme works as follows. First, a honeypot page achieves high ranking in a SERP by boosting techniques. But when the page is requested by a user, they don't actually see it, they get redirected to a target page. There are various ways to achieve redirection. The easiest approach is to set a page refresh time to zero and initialize a refresh URL attribute with a URL of a target page. More sophisticated approach is to use page level scripts that aren't usually executed by crawlers and hence more effective from spammers point of view.

- **Accessible Pages.** These are pages which spammers can modify but don't own. For instance, it can be Wikipedia pages, blog with public comments, a public discussion group, or even an open user-maintained web directory. Spammers exploit the opportunity to be able to slightly modify external pages by creating links to their own pages. It is worth noting that these strategies are usually combined. Thus, while spamming comments, adversaries can apply both link and anchor text spamming techniques.

2.3 Cloaking and Redirection

Cloaking is the way to provide different versions of a page to crawlers and users based on information contained in a request. If used with good motivation, it can even help search engine companies because in this case they don't need to parse a page in order to separate the core content from a noisy one (advertisements, navigational elements, rich GUI elements). However, if exploited by spammers, cloaking takes an abusive form. In this case adversary site owners serve different copies of a page to a crawler and a user with the goal to deceive the former [28; 108; 110; 75]. For example, a surrogate page can be served to the crawler to manipulate ranking, while users are served with a user-oriented version of a page. To distinguish users from crawlers spammers analyze a *user-agent* field of HTTP request and keep track of IP addresses used by search engine crawlers. The other strategy is to redirect users to malicious pages by executing JavaScript activated by page *onLoad()* event or timer. It is worth mentioning that JavaScript redirection spam is the most widespread and difficult to detect by crawlers, since mostly crawlers are script-agnostic [29].

2.4 Click Spam

Since search engines use click stream data as an implicit feedback to tune ranking functions, spammers are eager to generate fraudulent clicks with the intention to bias those functions towards their websites. To achieve this goal spammers submit queries to a search engine and then click on links pointing to their target pages [92; 37]. To hide anomalous behaviour they deploy click scripts on multiple machines or even in large botnets [34; 88]. The other incentive of spammers to generate fraudulent clicks comes from online advertising [60]. In this case, in reverse, spammers click on ads of competitors in order to decrease their budgets, make them zero, and place the ads on the same spot.

3. ALGORITHMS

All the algorithms proposed to date can be roughly categorized into three groups. The first one consists of techniques which analyze content features, such as word counts or language models [44; 85; 100; 81; 101; 89; 40], and content duplication [42; 43; 103]. Another group of algorithms utilizes link-based information such as neighbour graph connectivity [25; 47; 49; 48; 119; 118; 116], performs link-based trust and distrust propagation [86; 55; 71; 99; 112; 12; 52; 26; 66; 11; 5; 8], link pruning [16; 84; 73; 93; 74; 35; 109;

[7] now link farm maintenance is in dozens of times cheaper than before and the price is constantly decreasing

[8] some researchers consider it as a separate form of spamming technique

32; 111; 115; 6], graph-based label smoothing [121; 1; 30], and study statistical anomalies [4; 7; 38; 9]. Finally, the last group includes algorithms that exploit click stream data [92; 37; 60] and user behaviour data [77; 78], query popularity information [28; 10], and HTTP sessions information [107].

3.1 Content-based Spam Detection

Seminal line of work on content-based anti-spam algorithms has been done by Fetterly et al. [43; 44; 42; 85]. In [44] they propose that web spam pages can be identified through statistical analysis. Since spam pages are usually automatically generated, using phrase stitching and weaving techniques [54] and aren't intended for human visitors, they exhibit anomalous properties. Researchers found that the URLs of spam pages have exceptional number of dots, dashes, digits and length. They report that 80 of the 100 longest discovered host names refer to adult websites, while 11 refer to financial-credit-related websites. They also show that pages themselves have a duplicating nature – most spam pages that reside on the same host have very low word count variance. Another interesting observation is that the spam pages' content changes very rapidly[9]. Specifically, they studied average amount of week-to-week changes of all the web pages on a given host and found that the most volatile spam hosts can be detected with 97.2% based only on this feature. All the proposed features can be found in the paper [44].

In their other work [42; 43] they studied content duplication and found that the largest clusters with a duplicate content are spam. To find such clusters and duplicate content they apply shingling [22] method based on Rabin fingerprints [91; 21]. Specifically, they first fingerprint each of n words on a page using a primitive polynomial P_A, second they fingerprint each token from the first step with a different primitive polynomial P_B using prefix deletion and extension transformations, third they apply m different fingerprinting functions to each string from the second stage and retain the smallest of the n resulting values for each of the m fingerprinting functions. Finally, the document is represented as a bag of m fingerprints and clustering is performed by taking the transitive closure of the near-duplicate relationship[10]. They also mined the list of popular phrases by sorting (i, s, d) triplets[11] lexicographically and taking sufficiently long runs of triples with matching i and s values. Based on this study they conclude that starting from the 36^{th} position one can observe phrases that are evidence of machine-generated content. These phrases can be used as an additional input, parallel to common spam words, for a "bag of word"-based spam classifier.

In [85] they continue their analysis and provide a handful of other content-based features. Finally, all these features are blended in a classification model within C4.5, boosting, and bagging frameworks. They report 86.2% true positive and 97.8% true negative rates for a boosting of ten C4.5 trees. Recent work [40] describes a thorough study on how various features and machine learning models contribute to the quality of a web spam detection algorithm. The authors achieved superior classification results using state-of-the-art learning models, LogitBoost and RandomForest, and only cheap-to-compute content features. They also showed that computationally demanding and global features, for instance PageRank, yield only negligible additional increase in quality. Therefore, the authors claim that more careful and appropriate choice of a machine learning model is very important.

Another group introduces features based on HTML page structure to detect script-generated spam pages [103]. The underlying idea, that spam pages are machine-generated, is similar to the work discussed above [42; 43]. However, here authors make a non-traditional preprocessing step by *removing all the content and keeping only layout of a page.* Thus, they study page duplication by analyzing its layout and not content. They apply fingerprinting techniques [91; 21] with the subsequent clustering to find groups of structurally near-duplicate spam pages.

There is a line of work dedicated to language modelling for spam detection. [81] presents an approach of spam detection in blogs by comparing the language models [59] for blog comments and pages, linked from these comments. The underlying idea is that these models are likely to be substantially different for a blog and a spam page due to random nature of spam comments. They use KL-divergence as a measure of discrepancy between two language models (probability distributions) Θ_1, Θ_2:

$$KL(\Theta_1 \| \Theta_2) = \sum_w p(w|\Theta_1) \log \frac{p(w|\Theta_1)}{p(w|\Theta_2)}. \qquad (2)$$

The beneficial trait of this method is that it doesn't require any training data.

[101; 89] extend the analysis of linguistic features for web spam detection by considering lexical validity, lexical and content diversity, syntactical diversity and entropy, emotiveness, usage of passive and active voices, and various other NLP features.

Finally, [10] proposes a number of features based on occurrence of keywords on a page that are either of high advertising value or highly spammed. Authors investigate discriminative power of the following features: Online Commercial Intention (OCI) value assigned to a URL by Microsoft AdCenter[12], Yahoo! Mindset classification of a page as either commercial or non-commercial[13], Google AdWords popular keywords[14], and number of Google AdSense ads on a page[15]. They report an increase in accuracy by 3% over the [24] which doesn't use these features. Similar ideas were applied for cloaking detection. In [28] search engine query logs and online advertising click-through logs are analyzed with respect to query popularity and monetizability. Definition of query popularity is straightforward, query monetizability is defined as a revenue generated by all ads that are served in response to the query[16]. Authors use top 5000 queries out of each query category, request top 200 links for each query four times by providing various agent-fields to imitate requests by a user (u) and a crawler (c), and then apply a cloaking test (3), which is a modified version of a test pro-

[9] it can even change completely on every request.

[10] two documents are near-duplicates if their shingles agree in two out of six of the non-overlapping runs of fourteen shingles [42]

[11] s is the i^{th} shingle of document d

[12] adlab.msn.com/OCI/oci.aspx

[13] mindset.research.yahoo.com

[14] adwords.google.com/select/keywordtoolexternal

[15] google.com/adsense

[16] overlap between these two categories is reported to be 17%

posed in the earlier work on cloaking detection [108; 110]:

$$CloakingScore(p) = \frac{\min[D(c_1,u_1), D(c_2,u_2)]}{\max[D(c_1,c_2), D(u_1,u_2)]}, \quad (3)$$

where

$$D(a_1,a_2) = 1 - 2\frac{|a_1 \cap a_2|}{|a_1 \cup a_2|} \quad (4)$$

is a normalized term frequency difference for two copies of a page represented as a set of terms, may be with repetitions. They report 0.75 and 0.985 precision at high recall values for popular and monetizable queries correspondingly, which suggests that the proposed technique is useful for cloaking detection. However, the methods described in [28; 108; 110] might have relatively high false positive rate, since legitimate dynamically generated pages generally contain different terms and links on each access, too. To overcome this shortcoming an improved method was proposed [75], which is based on the analysis of structural (tags) properties of the page. The idea is to compare multisets of tags and not words and links of pages to compute the cloaking score.

3.2 Link-based Spam Detection

All link-based spam detection algorithms can be subdivided into five groups. The first group exploits the topological relationship (distance, co-citation, similarity) between the web pages and a set of pages for which labels are known [86; 55; 71; 99; 112; 12; 52; 26; 66; 11; 5; 8]. The second group of algorithms focuses on identification of suspicious nodes and links and their subsequent downweighting [16; 84; 73; 93; 74; 35; 109; 32; 111; 115; 6]. The third one works by extracting link-based features for each node and use various machine learning algorithms to perform spam detection [4; 7; 38; 9]. The fourth group of link-based algorithms uses the idea of labels refinement based on web graph topology, when preliminary labels predicted by the base algorithm are modified using propagation through the hyperlink graph or a stacked classifier [25; 47; 49; 48]. Finally, there is a group of algorithms which is based on graph regularization techniques for web spam detection [121; 1; 30].

3.2.1 Preliminaries

- **Web Graph Model.** We model the Web as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$, representing web pages[17], and directed weighted edges $\mathcal{E}$, representing hyperlinks between pages. If a web page p_i has multiple hyperlinks to a page p_j, we will collapse all these links into one edge $(i,j) \in \mathcal{E}$. Self loops aren't allowed. We denote a set of pages linked by a page p_i as $Out(p_i)$ and a set of pages pointing to p_i as $In(p_i)$. Finally, each edge $(i,j) \in \mathcal{E}$ can have an associated non-negative weight w_{ij}. A common strategy to assign weights is $w_{ij} = \frac{1}{|Out(p_i)|}$, though other strategies are possible. For instance, in [2] 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

$$M_{ij} = \begin{cases} w_{ij}, \text{ if } (i,j) \in \mathcal{E}, \\ 0, \quad \text{otherwise.} \end{cases} \quad (5)$$

[17] one can also analyze host level web graph

- **PageRank** [86] 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 p_i to a page p_j shows an endorsement or trust of page p_i in page p_j, 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

$$\vec{\pi} = (1-c) \cdot M^T \vec{\pi} + c \cdot \vec{r}, \quad (6)$$

where c is a *damping factor*, and $\vec{r}$ is a static PageRank vector. For non-personalized PageRank it is a unit vector $\left(\frac{1}{N}, \ldots, \frac{1}{N}\right)$, where $N = |\mathcal{V}|$. It is worth noting that this is needed to meet the conditions of a Perron-Frobenius theorem [15] and guarantee the existence of a stationary distribution for the corresponding Markov 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 $\vec{r}$ the solution is called a personalized PageRank (PPR) [46; 56] and the vector $\vec{r}$ is called a *personalization*, *random jump* or *teleportation* vector.

There are a few useful properties of PageRank. First, it is linear in $\vec{r}$, i.e. if $\vec{\pi}_1$ is the solution of (6) with the personalization vector $\vec{r}_1$ and $\vec{\pi}_2$ is the solution of (6) with the personalization vector $\vec{r}_2$, then the vector $\frac{\vec{\pi}_1 + \vec{\pi}_2}{2}$ is the solution for the equation (6) with the personalization vector $\frac{\vec{r}_1 + \vec{r}_2}{2}$. As a consequence of that we can expand a personalized PageRank in the following way:

$$PPR(\vec{r}) = \frac{1}{N} \sum_{v \in \mathcal{V}} PPR(\chi_v), \quad (7)$$

where χ_v is the teleportation vector consisting of all zeros except for a node v such that $\chi_v(v) = 1$ (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 [62; 45], i.e. the probability to make j steps before termination is equal to $c \cdot (1-c)^j$ and the following representation is valid

$$PPR(\vec{r}) = c\,\vec{r} \cdot \sum_{j=0}^{\infty} (1-c)^j (M^T)^j. \quad (8)$$

- **HITS** [68] algorithm assigns *hub* and *authority* scores for each page on the web and is based on the following observation. Page p_i is a good hub, has a high hub score h_i, 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 a_i. As 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:

$$\begin{cases} \vec{a} = A^T \vec{h}, \\ \vec{h} = A\vec{a}. \end{cases} \quad (9)$$

It can be shown that the solution $(\vec{h}, \vec{a})$ for the system of equations (9) after repeatitive updating converges to the principal eigenvectors of AA^T and A^TA correspondingly.

[83] 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 stable to small perturbations of a graph, while HITS is quite sensitive. [18] 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.

[53] studies optimality properties of link farms. 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 [72] and a survey on efficient PageRank computation [14].

3.2.2 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 [55], which propagates trust from a small seed set of good pages via a personalized PageRank. The algorithm 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 [71]. 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 BadRank [99], 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.

[112] further researches the 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 child[18] gets an equal part of parent's trust $c \cdot \frac{TR(p)}{|Out(p)|}$, and propose two more strategies:

- *constant splitting*, when each child gets the same discounted part of parent's trust $c \cdot TR(p)$ 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 $c \cdot \frac{TR(p)}{\log(1+|Out(p)|)}$.

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 parents scores. Finally, they propose to use a linear combination of trust and distrust values:

$$TotalScore(p) = \eta \cdot TR(p) - \beta \cdot AntiTR(p), \quad (10)$$

where η, $\beta \in (0, 1)$. 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 [51].

Two algorithms [12; 52] utilize PageRank decomposition property (Section *3.2.1*) to estimate the amount of undeserved PageRank coming from suspicious nodes. In [12] the SpamRank algorithm is proposed; it finds *supporters* for a page using Monte Carlo simulations [46], assigns a penalty score for each page by analyzing whether personalized PageRank score $PPR(\vec{\chi}_j)_i$ 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 [87]. The concept of *spam mass* was introduced in [52]. 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 $\vec{\pi}$ and TrustRank $\vec{\pi}'$ vectors and estimates the amount of spam mass for each page using the formula $\vec{m} = \frac{\vec{\pi} - \vec{\pi}'}{\vec{\pi}}$. 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.

[18] in this paragraph we will refer to out-neighbours of a page as children and in-neighbours as parents

Credibility-based link analysis is described in [26]. 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 *BadPath*, 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

$$C_k(p) = \left\{1 - \sum_{l=1}^{k}\Big[\sum_{path_l(p)\in BadPath_l(p)} P\big(path_l(p)\big)\Big]\right\}\cdot\gamma(p), \tag{11}$$

where k is a parameter specifying the length of a random walk, $\gamma(p)$ is a credibility penalty factor that is needed to deal with only partial knowledge of all spam pages on the Web[19], and $P\big(path_l(p)\big) = \prod_{i=0}^{l-1} w_{ii+1}$. The credibility score can be used to downweight or prune low credible links before link-based ranking or to change the personalization vector in PPR, TrustRank, or Anti-TrustRank.

In [66] 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; non-conserving rank, which is a generalization of personalized PageRank satisfying the equation

$$\vec{\pi} = \left(I - (1-c)\cdot M^T\right)^{-1}\vec{r}. \tag{12}$$

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 [11], where similarity-based top-K lists are used to compute a spam score for a new page. Authors consider co-citation, CompanionRank, SimRank [61], 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 [71] and combined Trust-Distrust [112] algorithms show higher precision at low recall levels.

The seminal line of work was done by Baeza-Yates et al. [5; 8; 9; 7]. In [5], 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

$$\vec{p} = \frac{1}{N}\vec{1}\sum_{j=0}^{\infty} damping(j)(M^T)^j, \tag{13}$$

[19]several strategies to define γ are proposed.

and prove the theorem that any damping function such that the sum of dampings 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 [8] 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

$$damping(j) = \begin{cases} 0 & \text{if } j \le J, \\ D(1-c)^j & \text{otherwise.} \end{cases} \tag{14}$$

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 [7; 9] correspondingly.

3.2.3 Link pruning and reweighting algorithms

Algorithms belonging to this category tend to find unreliable links and demote them. The seminal work [16] raises problems in HITS [68] algorithm, such as domination of mutually reinforcing relationships and neighbour 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 $\frac{1}{k}$ if there are k pages from one site link to a single page on another site, and assign a hub weight of $\frac{1}{l}$ 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 [84], 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 A^TA, 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 [73], which performs two random walks to estimate authority and hub scores for pages in a subgraph initially retrieved by keyword-based search. It is worth noting that the original and an inverted subgraphs are considered to get two different scores. An extension of this work [93] considers *clustering structure* on pages and their linkage patterns to downweight 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:

$$a_j = \sum_{k:\, j\in l(k)} \frac{1}{\sum_{i:\, j\in l(i)} S_{ik}}, \tag{15}$$

where $S_{ik} = \frac{|l(i)\cap l(k)|}{|l(i)\cup l(k)|}$ and $l(i)$ is a set of pages linked from

page p_i. The approach acts like popularity ranking methods discussed in [20; 27].

[74] 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.

[35] introduces the concept of "neponistic" links – links that present for reasons rather 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 downweighting neponistic links. In their other work [109] 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 neighbours 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 a lot of bad pages from the seed set. Finally, links between expanded set of bad pages are removed or downweighted and any link-based ranking algorithm [86; 68] can be applied. Similar ideas are studied on a host level in [32].

In [111] 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 $A_{ij} = 1$, if a page p_i contains complete-hyperlink chl_j. Then it finds bipartite components with the size greater than a threshold in the corresponding graph, where parts are pages and links, and downweight complete hyperlinks from large components. Finally, a HITS-like algorithm is applied on a reweighted adjacency matrix.

[115] 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 downweight outgoing links for them with some function proportional to correlation. They also prove that spammers can amplify PageRank score by at most $\frac{1}{c}$ and experimentally show that even two-node collusion can yield a big PageRank amplification. The follow-up work [6] performs more general analysis of different collusion topologies, where they show that due to the power law distribution of PageRank [87], the increase in PageRank is negligible for top-ranked pages. The work is similar to [53; 3].

3.2.4 Algorithms with link-based features

Algorithms from this category represent pages as feature vectors and perform standard classification or clustering analysis. [4] studies link-based features to perform website categorization based on their functionality. Their assumption is that sites sharing similar structural patterns, such as average page level or number of outlinks 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 [7]. Mixture of content-based and link-based features is used to combat against web spam in [38; 9], spam in blogs [69; 76].

3.2.5 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 [25] a few web graph-based refinement strategies are proposed. The algorithm in [25] works in two stages. First, labels are assigned using a spam detection algorithm discussed in [7]. Then, at the second stage labels are refined in one of three ways. One strategy is to perform Web graph clustering [67] 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 [120]. The key part is to initialize the personalization vector $\vec{r}$ in PPR by normalizing the predictions of the base algorithm: $r_p = \frac{s(p)}{\sum_{p \in \mathcal{V}} s(p)}$, where $s(p)$ is a prediction of the base algorithm and r_p is the component of the vector $\vec{r}$ corresponding to page p. Finally, the third strategy is to use stacked graphical learning [70]. 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 relabelling strategies are proposed in [47; 49; 48]. [47] 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. [48] proposes to use feature re-extraction strategy using clustering, propagation, and neighbour-graph analysis. Self-training was applied in [49] so as to reduce the size of the training dataset requiring for web spam detection.

3.2.6 Graph regularization algorithms

Algorithms in this group perform trunsductive 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 [121] builds a discrete analogue of classification regularization theory [102; 104] 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 $a_{ij} = \frac{w_{ji}}{In(p_i)}$. Second, they build the graph Laplacian

$$L = \Pi - \alpha \frac{\Pi A + A^T \Pi}{2}, \quad (16)$$

where α is a user-specified parameter in $[0, 1]$, A is a transition matrix, and Π is a diagonal matrix with the PageRank score[20] over the diagonal. Then, they solve the following matrix equation

$$L\vec{\varphi} = \Pi\vec{y}, \quad (17)$$

where $\vec{y}$ 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 $\vec{\varphi}$. It is worth noting that the algorithm requires strongly connected graphs.

Another algorithm that follows regularization theory is described in [1]. 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 [36; 27] 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:

$$\Omega(\vec{w}, \vec{z}) = \frac{1}{l}\sum_{i=1}^{l} L(\vec{w}^T\vec{x}_i + z_i, y_i) + \lambda_1\|\vec{w}\|^2 + \lambda_1\|\vec{z}\|^2 + \gamma \sum_{(i,j)\in\mathcal{E}} a_{ij}\Phi(\vec{w}^T\vec{x}_i + z_i, \vec{w}^T\vec{x}_j + z_j), \quad (18)$$

where $L(a, b)$ is a standard loss function, $\vec{w}$ is a vector of coefficients, $\vec{x}_i$ 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 $(i, j) \in \mathcal{E}$, and $\Phi(a, b) = \max[0, b - a]^2$ 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 [30]. 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), *post 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 (see eq. 5). 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.

[20] computed at the first stage.

3.3 Miscellaneous

In this section we discuss algorithms that use non-traditional features and ideas to combat against web spam.

3.3.1 Unsupervised spam detection

The problem of unsupervised web spam detection is studied in [119; 118; 116]. Authors propose the concept of *spamicity* and develop an online client-side algorithm for web spam detection. The key distinctive feature of their solution is that it doesn't need training data. At the core of their approach is a (θ, k)-*page farm model* introduced in [117], that allows to compute the theoretical bound on the PageRank score that can be achieved using the optimal page farm with a given number of pages and links between them. The proposed algorithm works as follows. For a given page it greedily select pages from the k-neighbourhood, that contribute most to the PageRank score, following the definition

$$PRContrib(v, p) = \begin{cases} PR[p, \mathcal{G}] - PR[p, \mathcal{G}(\mathcal{V} - \{v\})], \text{ if } v \neq p, \\ \frac{1-c}{N}, \quad \text{otherwise,} \end{cases} \quad (19)$$

and at each iteration it computes the link-spamicity score as the ratio of the observed PageRank contribution from selected pages over the optimal PageRank contribution. The algorithm uses monotonicity property to limit number of supporters that needs to be considered. Finally, it marks the page as suspicious if the entire k-neighbourhood of this page is processed and link-spamicity score is still greater than a threshold. Analogous optimality conditions were proposed for a page content. In this case term-spamicity score of a page is defined as the ratio of the TFIDF score achieved by the observed content of a page over the TFIDF score that could be achieved by an "optimal" page that has the same number of words.

3.3.2 Algorithms based on user browsing behaviour

[77] proposes to incorporate user browsing data for web spam detection. The idea is to build a *browsing graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \sigma)$ where nodes $\mathcal{V}$ are pages and edges $\mathcal{E}$ are transitions between pages, $\mathcal{T}$ corresponds to a staying time and σ denotes the random jump probability, and then compute an importance for each page using PageRank-like algorithm. The distinctive feature of the proposed solution is that it considers *continuous*-time Markov process as an underlying model because user browsing data include time information. Formally, the algorithm works as follows. First, using

staying times $Z_1, \ldots, Z_{m_i}$ for a page i it finds the diagonal element q_{ii} of the matrix $Q = P'(t)$ as a solution of the optimization problem:

$$\left[(\bar{Z} + \frac{1}{q_{ii}}) - \frac{1}{2}(S^2 - \frac{1}{q_{ii}^2})\right]^2 \longrightarrow \min_{q_{ii}},\ s.t.\ q_{ii} < 0. \quad (20)$$

Non-diagonal elements of the matrix Q are estimated as

$$-\frac{q_{ij}}{q_{ii}} = \begin{cases} c\frac{\tilde{w}_{ij}}{\sum_{k=1}^{N+1}\tilde{w}_{ij}} + (1-c)\sigma_j, & i \in \mathcal{V}, j \in \tilde{\mathcal{V}} \\ \sigma_j, & i = N+1, j \in \mathcal{V}, \end{cases} \quad (21)$$

where additional $N+1$ pseudo-vertex is added to the graph to model the teleportation effect such that all last pages in the corresponding browsing sessions are connected to it via edges with the weights equal to the number of clicks on the last page, and pseudo-vertex is connected to the first page in each session with the weight equal to normalized frequency of visits for this page; σ_j is a random jump probability. At the second step, the stationary distribution is computed using the matrix Q, which is proven to correspond to the stationary distribution for the continuous-time Markov process. Similar idea is studied in [90]. The authors define a *hyperlink-click* graph, which is a union of the standard Web graph and a query-page click-log-based, and apply random walks to perform pages ranking.

User behaviour data is also used in [78]. There are two observations behind their approach. First, spammers aim to get high ranking in SERP and therefore the major part of the traffic to spam website is due to visits from a search engine site. Second, users can easily recognize a spam site and therefore should leave it quite fast. Based on these observations authors propose 3 new features: ratio of visits from a search site, number of clicks and page views on a site per visit.

3.3.3 HTTP analysis and real-time spam detection

Methods in this subsection can be naturally partitioned into 2 groups: client-side and server-side. The methods from the first group use very limited information, usually doesn't require learning, and less accurate. The latter group representative methods, in reverse, are more precise since they can incorporate additional real-time information.

Lightweight client-side web spam detection method is proposed in [107]. Instead of analyzing content-based and link-based features for a page, researchers focused on HTTP session information and achieved competitive results. They represent each page and a session, associated with it, as a vector of features such as IP-address or words from a request header in a "bag-of-words" model, and perform a classification using various machine learning algorithms. The same group introduced the way of large dataset creation for web spam detection by extracting URLs from email spam messages [106]. Though not absolutely clean, the dataset contains about 350000 web spam pages.

Similarly, HTTP sessions are analyzed, among others, in the context of malicious redirection detection problem. The authors of [29] provide a comprehensive study of the problem, classify all spam redirection techniques into 3 types (HTTP redirection, META Refresh, JavaScript redirection), analyze the distribution of various redirection types on the Web, and present a lightweight method to detect JavaScript redirection, which is the most prevalent and difficult to identify type.

The idea to use rank-time features in addition to query-independent features is introduced in [100]. Specifically, authors solve the problem of spam pages demotion after the query was issued guided by the principle that spammers fool ranking algorithms and achieve high positions using different methods rather than genuinely relevant pages, and therefore spam pages should be outliers. Overall, 344 rank-time features are used such as number of query terms in title and frequency of a query term on a page, number of pages that contain a query term, n-gram overlaps between query terms and a page, for different values of n and for different *skip n-grams*. According to the experiment, the addition of rank-time features allows to increase precision by 25% at the same levels of recall. In the same work they study the problem of overfitting in web spam detection and suggest that training and testing data should be domain-separated, otherwise testing error could be up to 40% smaller than the real.

3.3.4 Click spam detection

Since click spam aims to push "*malicious noise*" into a query log with the intention to corrupt data, used for the ranking function construction, most of the counter methods study the ways to make learning algorithms robust to this noise. Other anti-click-fraud methods are driven by the analysis of the economic factors underlying the spammers ecosystem.

Interesting idea to prevent click spam is proposed in [92]. The author suggests using personalized ranking functions, as being more robust, to prevent click fraud manipulation. The paper presents a utility-based framework allowing to judge when it is economically reasonable to hire spammers to promote a website, and performs experimental study demonstrating that personalized ranking is resistant to spammers manipulations and diminishes financial incentives of site owners to hire spammers. The work [37] studies the robustness of the standard click-through-based ranking function construction process and also reports its resistance to fraudulent clicks.

The work [60] studies the problem of click fraud for online advertising platform and particularly addresses the problem of "competitor bankruptcy". The authors present a *click-based* family of ads pricing models and theoretically prove that such models leave no economic incentives for spammers to perform malicious activity, i.e. short term competitor's budget wasting will be annihilated by long term decrease in the ads placement price. [105] carefully analyzes the entire spammers' ecosystem by proposing the spam double funnel model which describes the interaction between spam-publishers and advertisers via page redirections.

In [17] an incentive based ranking model is introduced, which mainly incorporates users into ranking construction and provides a profitable arbitrage opportunity for the users to correct inaccuracies in the system. The key idea is that users are subject to an explicit information about revenue they might "earn" within the system if they correct an erroneous ranking. It is theoretically proven that the model with the specific incentives (revenue) structure guarantees merit-based ranking and is resistant to spam.

4. KEY PRINCIPLES

Having analyzed all the related work devoted to the topic of web spam mining, we identify a set of underlying principles that are frequently used for algorithms construction.

- Due to machine-generated nature and its focus on search engines manipulation, spam shows abnormal properties such as high level of duplicate content and links; rapid changes of content; and the language models built for spam pages deviate significantly from the models built for the normal Web.
- Spam pages deviate from power law distributions based on numerous web graph statistics such as PageRank or number of in-links.
- Spammers mostly target popular queries and queries with high advertising value.
- Spammers build their link farms with the aim to boost ranking as high as possible, and therefore link farms have specific topologies that can be theoretically analyzed on optimality.
- According to experiments, the principle of approximate isolation of good pages takes place: good pages mostly link to good pages, while bad pages link either to good pages or a few selected spam target pages. It has also been observed that connected pages have some level of semantic similarity – topical locality of the Web, and therefore label smoothing using the Web graph is a useful strategy.
- Numerous algorithms use the idea of trust and distrust propagation using various similarity measures, propagation strategies and seed selection heuristics.
- Due to abundance of "neponistic" links, that negatively affect the performance of a link mining algorithm, there is a popular idea of links removal and downweighting. Moreover, the major support is caused by the k-hop neighbourhood and hence it makes sense to analyze local subgraphs rather than the entire Web graph.
- Because one spammer can have a lot of pages under one website and use them all to boost ranking of some target pages, it makes sense to analyze host graph or even perform clustering and consider clusters as a logical unit of link support.
- In addition to traditional page content and links, there are a lot of other sources of information such as user behaviour or HTTP requests. We hope that more will be developed in the near future. Clever feature engineering is especially important for web spam detection.
- Despite the fact that new and sophisticated features can boost the state-of-the-art further, proper selection and training of a machine learning models is also of high importance.

5. CONCLUSIONS

In this work we surveyed existing techniques and algorithms created to fight against web spam. To draw a general picture of the web spam phenomenon, we first provide numeric estimates of spam on the Web, discuss how spam affects users and search engine companies, and motivate academic research. We also presented a brief overview of various spam forms to make this paper self-contained and comprehensible to a broad audience of specialists. Then we turn to the discussion of numerous algorithms for web spam detection, and analyze their characteristics and underlying ideas. At the end, we summarize all the key principles behind anti-spam algorithms.

According to this work, web spam detection research has gone through a few generations: starting from simple content-based methods to approaches using sophisticated link mining and user behaviour mining techniques. 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 even more exciting and effective methods will be developed in the future.

Among promising directions of research we identify click-fraud for online advertising detection and construction of platforms, which don't have incentives for non-fair behaviour. For instance, *pay-per-click* models having this property will be very beneficial. Dynamic malicious redirection and detection of cloaking are still open issues. We also see the potential and need in anti-spam methods at the intersection of Web and social media.

6. REFERENCES

[1] J. Abernethy, O. Chapelle, and C. Castillo. Graph regularization methods for web spam detection. *Mach. Learn.*, Vol. 81, Nov. 2010.

[2] J. Abernethy, O. Chapelle, C. Castillo, J. Abernethy, O. Chapelle, and C. Castillo. WITCH: A new approach to web spam detection. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'08, 2008.

[3] S. Adali, T. Liu, and M. Magdon-Ismail. Optimal Link Bombs are Uncoordinated. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'05, Chiba, Japan, 2005.

[4] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. The connectivity sonar: detecting site functionality by structural patterns. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, Nottingham, UK, 2003.

[5] R. Baeza-Yates, P. Boldi, and C. Castillo. Generalizing pagerank: damping functions for link-based ranking algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'06, Seattle, Washington, 2006.

[6] R. Baeza-Yates, C. Castillo, and V. López. Pagerank Increase under Different Collusion Topologies. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'05, 2005.

[7] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of web spam. In *Proceedings of the Second In-*

ternational Workshop on Adversarial Information Retrieval on the Web, AIRWeb'06, Seattle, USA, 2006.

[8] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Using rank propagation and probabilistic counting for link-based spam detection. In *Proceedings of the Workshop on Web Mining and Web Usage Analysis*, WebKDD'06, Philadelphia, USA, 2006.

[9] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Web spam detection: Link-based and content-based techniques. In *The European Integrated Project Dynamically Evolving, Large Scale Information Systems (DELIS): proceedings of the final workshop*, volume Vol. 222, 2008.

[10] A. Benczúr, I. Bíró, K. Csalogány, and T. Sarlós. Web spam detection via commercial intent analysis. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'07.

[11] A. Benczúr, K. Csalogány, and T. Sarlós. Link-based similarity search to fight Web spam. In *Proceedings of the Second Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'06, Seattle, WA, 2006.

[12] A. A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher. Spamrank: Fully automatic link spam detection work in progress. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'05, May 2005.

[13] A. A. Benczúr, D. Siklósi, J. Szabó, I. Bíró, Z. Fekete, M. Kurucz, A. Pereszlényi, S. Rácz, and A. Szabó. Web spam: a survey with vision for the archivist. In *Proceedings of the International Web Archiving Workshop*, IWAW'08.

[14] P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, Vol. 2, 2005.

[15] A. Berman and R. Plemmons. *Nonnegative Matrices in the Mathematical Sciences (Classics in Applied Mathematics)*. Society for Industrial Mathematics, 1987.

[16] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'98, Melbourne, Australia.

[17] R. Bhattacharjee and A. Goel. Algorithms and Incentives for Robust Ranking. Technical report, Stanford University, 2006.

[18] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Trans. Internet Technol.*, Vol. 5, Feb. 2005.

[19] E. Blanzieri and A. Bryl. A survey of learning-based techniques of email spam filtering. *Artif. Intell. Rev.*, 29, March 2008.

[20] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW'01, Hong Kong, 2001.

[21] A. Z. Broder. Some applications of rabin's fingerprinting method. In *Sequences II: Methods in Communications, Security, and Computer Science*. Springer-Verlag, 1993.

[22] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Selected papers from the Sixth International Conference on World Wide Web*, WWW'97.

[23] C. Castillo and B. D. Davison. Adversarial web search. *Found. Trends Inf. Retr.*, 4, May 2011.

[24] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. A reference collection for web spam. *SIGIR Forum*, 40, Dec. 2006.

[25] 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 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'07, Amsterdam, The Netherlands, 2007.

[26] J. Caverlee and L. Liu. Countering web spam with credibility-based link analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, PODC'07, Portland, OR.

[27] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2002.

[28] K. Chellapilla and D. Chickering. Improving cloaking detection using search query popularity and monetizability, 2006.

[29] K. Chellapilla and A. Maykov. A taxonomy of javascript redirection spam. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb'07, Banff, Canada, 2007.

[30] Z. Cheng, B. Gao, C. Sun, Y. Jiang, and T.-Y. Liu. Let web spammers expose themselves. In *Proceedings of the fourth ACM International Conference on Web search and Data Mining*, WSDM'11, Hong Kong, China, 2011.

[31] E. Convey. Porn sneaks way back on web. *The Boston Herald*, 1996.

[32] A. L. da Costa Carvalho, P. A. Chirita, E. S. de Moura, P. Calado, and W. Nejdl. Site level noise removal for search engines. In *Proceedings of the 15th International Conference on World Wide Web*, WWW'06, Edinburgh, Scotland, 2006.

[33] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD'04, WA, USA, 2004.

[34] N. Daswani and M. Stoppelman. The anatomy of clickbot.a. In *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, Berkeley, CA, 2007. USENIX Association.

[35] B. Davison. Recognizing nepotistic links on the web. In *Workshop on Artificial Intelligence for Web Search*, AAAI'00.

[36] B. D. Davison. Topical locality in the web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'00, Athens, Greece.

[37] Z. Dou, R. Song, X. Yuan, and J.-R. Wen. Are clickthrough data adequate for learning web search rankings? In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM'08, 2008.

[38] I. Drost and T. Scheffer. Thwarting the nigritude ultramarine: Learning to identify link spam. In *Proceedig of the 16th European Conference on Machine Learning*, ECML'05, 2005.

[39] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the web frontier. In *Proceedings of the 13th International Conference on World Wide Web*, WWW'04, New York, NY, 2004.

[40] M. Erdélyi, A. Garzó, and A. A. Benczúr. Web spam classification: a few features worth more. In *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality*, WebQuality'11, Hyderabad, India, 2011.

[41] D. Fetterly. *Adversarial Information Retrieval: The Manipulation of Web Content*. 2007.

[42] D. Fetterly, M. Manasse, and M. Najork. Detecting phrase-level duplication on the world wide web. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'05, Salvador, Brazil.

[43] D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. *J. Web Eng.*, 2, Oct. 2003.

[44] 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: colocated with ACM SIGMOD/PODS 2004*, WebDB'04, Paris, France, 2004.

[45] D. Fogaras. Where to start browsing the web. In *Proceedings of IICS*. Springer-Verlag, 2003.

[46] D. Fogaras and B. Racz. Towards scaling fully personalized pagerank. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph*, WAW'04, 2004.

[47] Q. Gan and T. Suel. Improving web spam classifiers using link structure. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'07, Banff, Alberta, 2007.

[48] G. Geng, C. Wang, and Q. Li. Improving web spam detection with re-extracted features. In *Proceeding of the 17th International Conference on World Wide Web*, WWW'08, Beijing, China, 2008.

[49] G.-G. Geng, Q. Li, and X. Zhang. Link based small sample learning for web spam detection. In *Proceedings of the 18th international conference on World Wide Web*, WWW'09, Madrid, Spain, 2009.

[50] googleblog.blogspot.com. http://googleblog.blogspot.com/2011/01/google-search-and-search-engine-spam.html,2011.

[51] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, WWW'04, New York, NY, 2004.

[52] Z. Gyongyi and H. Garcia-Molina. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Databases*, VLDB'06.

[53] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB'05, Trondheim, Norway, 2005. VLDB Endowment.

[54] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *Proceeding of the First International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'05, Chiba, Japan, May 2005.

[55] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, VLDB'04, Toronto, Canada, 2004.

[56] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web*, WWW'02, 2002.

[57] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36, 2002.

[58] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, Vol. 11(6), Nov. 2007.

[59] D. Hiemstra. Language models. In *Encyclopedia of Database Systems*. 2009.

[60] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar. Click Fraud Resistant Methods for Learning Click-Through Rates. Technical report, Microsoft Research, Redmond, 2006.

[61] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD'02, Edmonton, Alberta, 2002.

[62] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, WWW'03, Budapest, Hungary, 2003.

[63] R. Jennings. The global economic impact of spam. *Ferris Research*, 2005.

[64] R. Jennings. Cost of spam is flattening – our 2009 predictions. *Ferris Research*, 2009.

[65] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'05, Salvador, Brazil, 2005.

[66] A. Joshi, R. Kumar, B. Reed, and A. Tomkins. Anchor-based proximity measures. In *Proceedings of the 16th International Conference on World Wide Web*, WWW'07, Banff, Alberta, 2007.

[67] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, Vol. 48, 1998.

[68] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46, Sept. 1999.

[69] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: a machine learning approach. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume Vol. 2, Boston, MA, 2006. AAAI Press.

[70] Z. Kou and W. W. Cohen. Stacked graphical models for efficient inference in markov random fields. In *Proceedings of the Seventh SIAM International Conference on Data Mining*, SDM'07, Minneapolis, Minnesota, April 2007.

[71] V. Krishnan and R. Raj. Web spam detection with anti-trust rank, 2006.

[72] A. Langville and C. Meyer. Deeper inside pagerank. *Internet Mathematics*, Vol. 1, 2004.

[73] R. Lempel and S. Moran. SALSA: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, 19, April 2001.

[74] L. Li, Y. Shang, and W. Zhang. Improvement of hits-based algorithms on web documents. In *Proceedings of the 11th International Conference on World Wide Web*, WWW'02, Honolulu, Hawaii, 2002.

[75] J.-L. Lin. Detection of cloaked web spam by using tag-based methods. *Expert Syst. Appl.*, 36, May 2009.

[76] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'07, Banff, Alberta, 2007.

[77] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. 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, Singapore, 2008.

[78] Y. Liu, M. Zhang, S. Ma, and L. Ru. User behavior oriented web spam detection. In *Proceeding of the 17th International Conference on World Wide Web*, WWW'08, Beijing, China, 2008.

[79] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, 2008.

[80] O. A. Mcbryan. GENVL and WWWW: Tools for taming the web. In *Proceedings of the First World Wide Web Conference*, WWW'94, Geneva, Switzerland, May 1994.

[81] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'05, Chiba, Japan, May 2005.

[82] M. Najork. Web spam detection, 2006.

[83] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors and stability. In *Proceedings of the 17th International Joint Conference on Artificial intelligence*, Seattle, WA, 2001. Morgan Kaufmann Publishers Inc.

[84] S. Nomura, S. Oyama, T. Hayamizu, and T. Ishida. Analysis and improvement of hits algorithm for detecting web communities. *Syst. Comput. Japan*, 35, Nov. 2004.

[85] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on World Wide Web*, WWW'06, Edinburgh, Scotland, 2006.

[86] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[87] G. Pandurangan, P. Raghavan, and E. Upfal. Using pagerank to characterize web structure. In *Proceedings of the 8th Annual International Conference on Computing and Combinatorics*, COCOON'02, London, UK, 2002. Springer-Verlag.

[88] Y. Peng, L. Zhang, J. M. Chang, and Y. Guan. An effective method for combating malicious scripts clickbots. In *Proceedings of the 14th European Conference on Research in Computer Security*, ESORICS'09, Berlin, Heidelberg, 2009.

[89] J. Piskorski, M. Sydow, and D. Weiss. Exploring linguistic features for web spam detection: a preliminary study. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'08, Beijing, China.

[90] B. Poblete, C. Castillo, and A. Gionis. Dr. searcher and mr. browser: a unified hyperlink-click graph. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM'08, 2008.

[91] M. Rabin. Fingerprinting by Random Polynomials. Technical report, Center for Research in Computing Technology, Harvard University, 1981.

[92] F. Radlinski. Addressing malicious noise in clickthrough data. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb'07, Banff, Canada, 2007.

[93] G. Roberts and J. Rosenthal. Downweighting tightly knit communities in World Wide Web rankings. *Advances and Applications in Statistics (ADAS)*, 2003.

[94] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM'04, Washington, D.C., 2004.

[95] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. AAAI'98, Madison, Wisconsin, July 1998.

[96] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, Vol.18, Nov. 1975.

[97] searchengineland.com. http://searchengineland.com/businessweek-dives-deep-into-googles-search-quality-27317, 2011.

[98] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33, Sept. 1999.

[99] M. Sobek. Pr0 - google's pagerank 0 penalty. badrank. http://pr.efactory.de/e-pr0.shtml, 2002.

[100] K. M. Svore, Q. Wu, C. J. C. Burges, and A. Raman. Improving web spam classification using rank-time features. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'07, Banff, Alberta, 2007.

[101] M. Sydow, J. Piskorski, D. Weiss, and C. Castillo. Application of machine learning in combating web spam, 2007.

[102] A. Tikhonov and V.Arsenin. Solutions of ill-posed problems, 1977.

[103] T. Urvoy, T. Lavergne, and P. Filoche. Tracking Web Spam with Hidden Style Similarity. In *Proceedings of the Second International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'06, Seattle, Washington, Aug. 2006.

[104] G. Wahba. Spline models for observational data. *CBMS-NSF Regional Conference Series in Applied Mathematics*, Vol. 59, 1990.

[105] Y.-M. Wang, M. Ma, Y. Niu, and H. Chen. Spam double-funnel: connecting web spammers with advertisers. In *Proceedings of the 16th International Conference on World Wide Web*, WWW'07, Banff, Alberta.

[106] S. Webb, J. Caverlee, and C. Pu. Characterizing web spam using content and HTTP session analysis. In *Proceedings of CEAS*, 2007.

[107] S. Webb, J. Caverlee, and C. Pu. Predicting web spam with HTTP session information. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, CIKM'08, 2008.

[108] B. Wu and B. Davison. Cloaking and redirection: A preliminary study, 2005.

[109] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Special interest tracks and posters of the 14th International Conference on World Wide Web*, WWW'05, Chiba, Japan, 2005.

[110] B. Wu and B. D. Davison. Detecting semantic cloaking on the web. In *Proceedings of the 15th International Conference on World Wide Web*, WWW'06, Edinburgh, Scotland, 2006.

[111] B. Wu and B. D. Davison. Undue influence: eliminating the impact of link plagiarism on web search rankings. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC'06, Dijon, France, 2006.

[112] B. Wu, V. Goel, and B. D. Davison. Propagating trust and distrust to demote web spam. In *Proceedings of the Workshop on Models of Trust for the Web*, Edinburgh, Scotland, May 2006.

[113] K. Yoshida, F. Adachi, T. Washio, H. Motoda, T. Homma, A. Nakashima, H. Fujikawa, and K. Yamazaki. Density-based spam detector. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'04.

[114] C. Zhai. *Statistical Language Models for Information Retrieval*. Now Publishers Inc., Hanover, MA, 2008.

[115] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. Van Roy. *Making Eigenvector-Based Reputation Systems Robust to Collusion*. LNCS Vol. 3243. Springer Berlin, Heidelberg, 2004.

[116] B. Zhou and J. Pei. OSD: An online web spam detection system. In *In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'09, Paris, France.

[117] B. Zhou and J. Pei. Sketching landscapes of page farms. In *Proceedings of the SIAM International Conference on Data Mining*, SDM'07, April.

[118] B. Zhou and J. Pei. Link spam target detection using page farms. *ACM Trans. Knowl. Discov. Data*, 3, July 2009.

[119] B. Zhou, J. Pei, and Z. Tang. A spamicity approach to web spam detection. In *Proceedings of the SIAM International Conference on Data Mining*, SDM'08, Atlanta, Georgia, April 2008.

[120] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, and B. S. Olkopf. Learning with Local and Global Consistency. In *Proceedings of the Advances in Neural Information Processing Systems 16*, volume Vol. 16, 2003.

[121] D. Zhou, C. J. C. Burges, and T. Tao. Transductive link spam detection. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb'07, Banff, Alberta, 2007.