Deeply Exploiting Link Structure: Setting a Tougher Life for Spammers

Yan Zhang Qiancheng Jiang Lei Zhang Yizhen Zhu
Department of Machine Intelligence, Peking University, Beijing 100871, China
{zhy, jqc, zhangl, zhuyz}@cis.pku.edu.cn

ABSTRACT

Previous anti-spamming algorithms based on link structure suffer from either the weakness of the page value metric or the vagueness of the seed selection. In this paper, we propose two page value metrics, AVRank and HVRank. These two “values” of all the web pages can be well assessed by using the bidirectional links’ information. Moreover, with the help of bidirectional links, it becomes easier to enlarge the propagation coverage and reduce the bias of seed sets. We further discuss the effectiveness of the combination of these two metrics, such as the quadratic mean of them. Our experimental results show that with such two metrics, an automatically selected large seed set can achieve a better propagation coverage as well as less bias of ranking results. Most important, our method can filter out spam sites and identify reputable sites more effectively.

1. INTRODUCTION

1.1 Motivation

Since their first emergence in the year of 1994, search engines have been more and more dominant in Web information discovery and retrieval. Therefore, a superior ranking position in search results becomes crucial for web sites to be prevalent, especially those commercial ones whose popularity leads to revenue. Thus many dishonest sites use deceptive methods to boost their ranking positions. Link spamming is the most nasty adversary among all these spamming tricks. For example, spammers build link farms [14] to achieve higher PageRank values for target pages or create pages solely to mislead search engines. They also attempt to get incoming links from reputable pages which can improve their rankings effectively.

To combat these rampant link spamming tricks, many link-based anti-spamming techniques have been proposed so far. TrustRank [15] assumes that good web pages seldom point to bad ones. Therefore good pages can propagate their trust via their outgoing links. 

AVRank outperforms traditional link-based algorithms such as PageRank on identifying good pages. On the other hand, some approaches hold the philosophy that a page should be penalized for pointing to bad pages. Parent Penalty [26] and Anti-Trust Rank [19] are typical methods of this kind. However, all of these methods only employ one type of incoming and outgoing links. Besides, they just deliver a single score to judge a page’s value. Inspired by HITS algorithm [18], we propose two generalized metrics, AVRank and HVRank, to measure a page’s value simultaneously. AVRank of a page is determined by its parent pages’ AVRank and HVRank scores, while HVRank of a page is determined by its child pages’ AVRank and HVRank scores. Our experimental results show that AVRank and HVRank can be used independently to detect spam pages effectively. More excited, the combination of AVRank and HVRank (e.g., the quadratic mean of them) is much better than either of them.

The other task of our work is seed selection. Previous methods such as TrustRank and Parent Penalty are all kinds of biased PageRank. Thus the seed set is crucial for them. To the best of our knowledge, seed selection is a manual and time-consuming process. Moreover, none of these methods has taken the propagation coverage and result bias of the seed set into consideration, which are critical to the final results.

In our work, we discover that the propagation ability of a seed set has a great relationship with web graph structure [20, 2] and the propagating direction. Web graph is an extremely sparse graph in nature which can be divided into several parts. Seeds in different parts have different propagation characteristics. In previous work such as TrustRank [15], the seed set must be carefully selected from different parts in order to maximize the propagation coverage. In our experiment, two different seed sets will be exploited and evaluated. One is a carefully selected small seed set (named as X) and the other is an automatically selected large seed set (named as Z). We demonstrate that the large seed set Z performs better in propagation coverage and result bias avoidance than the small seed set X. Furthermore, although the quantity of seeds increases, Z is time saving because it can be selected automatically other than manually.

1.2 Our Contributions

This paper mainly have three contributions to spamming detection algorithms. They are listed as follows:

- First, we propose a novel algorithm which considers both authority value and hub value of a page. This method makes use of bidirectional links and web structure to combat web spamming. It can better filter out spam pages and overcome the deficiency of the existing spam detection algorithms to some extent.
exploit the link dependencies among Web pages and the content of the pages themselves as features to build a decision tree classifier. They use machine learning techniques to promote the F-measure of the classification.

Haixuan Yang et al. [29] propose a ranking algorithm called DiffusionRank. It is motivated by the heat diffusion model. They consider the links among the pages as heat flow and give a formulation of their algorithm. It is a generalization of the PageRank which is immune to the link farms.

There are also some researches on content-based spam detection. Alexandros Ntoulas et al. [23] use statistical methods to many page content features to filter out spam pages.

3. EXPLOITING BIDIRECTIONAL LINKS

3.1 Assessing Page Value

Inspired by HITS algorithm [6, 18], there are two types of highly valuable pages: authorities and hubs. Authorities have plenty of good incoming links while hubs have plenty of good outgoing links.

Therefore, instead of using a single score to measure a page’s value, we evaluate a page’s authority value and hub value simultaneously. These two metrics are AVRank (AV for Authority Value) and HVRank (HV for Hub Value), which can be regarded as the generalization of authority score and hub score in HITS algorithm.

For a page $p$, $AVRank(p)$ is determined by its parent pages’ authority value and hub value, while $HVRank(p)$ is determined by its child pages’ authority value and hub value. As $p$ is pointed by lots of good pages, it will obtain a high $AVRank$ score. Similarly, page $p$ will obtain a high $HVRank$ score if it points to lots of good pages.

As PageRank and HITS algorithms, the scores of these two metrics are split equally by the number of links. Accordingly, the two metrics can be formalized as follows:

$$AVRank(p) = \sum_{q_i: \text{points to } p} \left( \alpha \cdot \frac{AVRank(q_i)}{ON(q_i)} + (1 - \alpha) \cdot \frac{HVRank(q_i)}{ON(q_i)} \right)$$

$$HVRank(p) = \sum_{r_i: \text{pinned by } p} \left( \beta \cdot \frac{AVRank(r_i)}{IN(r_i)} + (1 - \beta) \cdot \frac{HVRank(r_i)}{IN(r_i)} \right),$$

where $ON(q_i)$ is the outlink number of page $q_i$, and $IN(r_i)$ is the inlink number of page $r_i$. The parameters $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ in the formulas are used to denote the ratios of each part.

The equivalent matrix equation form is:

$$AVRank = \alpha M^T \cdot AVRank + (1 - \alpha) M^T \cdot HVRank$$

$$HVRank = \beta N^T \cdot AVRank + (1 - \beta) N^T \cdot HVRank$$

where $M$ is the line-normalized Web graph matrix, $N$ is the line-normalized inverse Web graph matrix, i.e., reverse every edge’s direction of Web graph. Both $M^T$ and $N^T$ are column-stochastic matrices.

The $AVRank$ and $HVRank$ are regressed to the authority and hub scores in HITS when assigning $\alpha = 0$ and $\beta = 1$ in Eq.(3.1) and Eq.(3.2). Therefore the HITS algorithm is a special case of ours. The proof of the computation convergence of the $AVRank$ and $HVRank$ formula is given in Appendix A.

3.2 Walking along the “Back-and-Forth” Surfer Model

It is well known that the PageRank has a good explanation through the “random surfer” model[24]. Web users randomly jump from

• Seed set is a crucial factor for these biased ranking algorithms. We discover that a large seed set not only increases the coverage, but also avoids the bias of the ranking results towards seeds. Furthermore, the process of our seed selection is automatic and time saving. It proves to be an effective method in our experiment.

• We experimentally show that our method with a large seed set can improve ranking results by maintaining reputable sites’ positions and demoting spam sites’ positions simultaneously.

The remainder of the paper is structured as follows: Section 2 gives an overview of related works. Section 3 introduces the idea of our method, the formula and explanation of the visiting model. Section 4 discusses seed selection and propagation coverage of seed sets. Section 5 demonstrates our experimental results. The last section concludes our work.

2. RELATED WORK

Web spamming receives a lot of attention recently [1, 3, 4, 10, 13, 15, 19, 22, 26, 27, 11]. Henzinger et al. identify web spamming as one of the biggest challenges to web search engines[16].

Page et al. first introduce PageRank algorithm which is the most popular method used by many search engines [24]. Bianchini et al. and Langville et al. analyze the PageRank mathematically [5, 21].

Kleinberg first discusses HITS algorithm which gives both authority score and hub score to a page [18]. Borodin et al. summarize early link-based ranking algorithms such as PageRank, SALSA and etc. [6]. R. Kumar et al. [20, 2] give an analysis of web graph structure. Web graph mainly consists of the following components: MAIN, IN, OUT, Tendrils and ISLANDS.

Gyongyi et al. summarize several spamming techniques used by spammers, such as term spamming, link spamming and hiding techniques [14]. They also illustrate many conventional forms of artificially inflating link popularity, with a particular focus on the inflation of PageRank[15].

Gyongyi et al. propose TrustRank algorithm to combat Web spam[15]. They first select a set of known trusted sites manually as seeds and assign high trust scores to them. Then trust scores are propagated via outlinks to other pages. However, they do not discuss the propagation coverage of the selected seeds. Therefore the efficiency of TrustRank cannot be guaranteed. In their following research [12], they introduce the concept of spam mass and discuss how to estimate spam mass and how to use the estimation to identify spam pages. Baoning Wu et al. make an improvement of TrustRank by using topical information in their work [28]. Topical TrustRank can decrease spam by 19% ~ 43.1% from top ranked sites when compared with TrustRank.

Baoning Wu et al. also propose an approach based on propagating negative value among pages with more strict rules[26]. Later they propose an approach by combining trust and distrust [27]. They compute trust score and distrust score separately and get a single score by a linear combination of these two scores. However, in their research, trust score has no relationship with distrust score.

Metaxas et al. discuss an anti-propagandistic method for anti-spamming[22]. It starts with an untrustworthy site $s$, then expands a graph by backlinks. They focus on the biconnected component (BCC) that includes $s$. Those pages that have at least two different paths to $s$ are also regarded as untrustworthy pages.

Luca Becchetti et al. [3, 4] use link-based characteristics and attributes to build a classifier automatically. These attributes are including indegree, outdegree, edge-reciprocity, PageRank, TrustRank, Truncated Rank and etc.. This classifier gains a high precision and low false positives in their experiments. Castillo et al. [7]
one page to a new page by following a hyperlink. However, this model cannot depict most of users’ browsing behavior nowadays. Instead of following hyperlinks, users often bounce back with “back button” on browser and choose a new page randomly. Marcin Sydow uses this “random surfer with back step” model [25] to improve the PageRank algorithm. It is called “back-and-forth” model in our research.

Our new metrics can well match the “back-and-forth” model. According to formula (A.4) in Appendix A:

\[
\text{AVRank} = \alpha \mathbf{M}^T + (1 - \alpha) \beta \mathbf{M}^T \cdot \mathbf{N}^T + (1 - \alpha)(1 - \beta)\mathbf{M}^T \cdot (\mathbf{N}^T)^2 + \ldots \cdot \text{AVRank}
\]

The first part \(\alpha \mathbf{M}^T\) corresponds to random visit. It happens at the probability of \(\alpha\). The second part is \((1 - \alpha) \beta \mathbf{M}^T \cdot \mathbf{N}^T\). The product of matrix \(\mathbf{M}^T\) and \(\mathbf{N}^T\) is still a column-stochastic matrix and the element \(M^T \cdot N^T[i, j]\) is just the transition probability from page \(p_i\) to \(p_j\) by first backing to previous page from page \(p_i\) and then clicking page \(p_j\). According to “back-and-forth” surfer model, the probability of this kind of visit is \((1 - \alpha) \beta\). The remaining parts can be explained in the same way with the “back-and-forth” model.

3.3 CPV Algorithm: Computing Page Values

Our CPV (Computing Page Values) algorithm is shown in Figure 1. It computes AVRank and HVRank scores for a web graph. The inputs of the algorithm are the web graph (matrices \(\mathbf{M}, \mathbf{N}\) and \(\mathbf{n}\) web pages), initial values and parameters that control algorithm execution. During each iteration, scores are attenuated and refilled.

```
input
\(\mathbf{M}\) transition matrix, line normalization
\(\mathbf{N}\) transpose of transition matrix, line normalization
AVRank\(_i\) initial authority values
LVRank\(_i\) initial hub values
\(\alpha, \beta\) weight of authority and hub
\(\alpha_d\) attenuation factor
\(M\) number of iterations
output
AVRank authority values
HVRank hub values
begin
AVRank = AVRank\(_i\)
HVRank = LVRank\(_i\)
for \(i = 1\) to \(M\)
\(\text{AVRank} = \alpha \cdot \mathbf{M}^T \cdot \text{AVRank} + (1 - \alpha) \cdot \mathbf{M}^T \cdot \text{HVRank}\)
\(\text{HVRank} = \beta \cdot \mathbf{N}^T \cdot \text{AVRank} + (1 - \beta) \cdot \mathbf{N}^T \cdot \text{HVRank}\)
end for
return AVRank, HVRank
end
```

Figure 1: CPV algorithm

![Figure 1: CPV algorithm](image1)

The CPV algorithm can start from uniform initial values (without seeds) or personalized initial values (with seeds). In order to illustrate our algorithm, a toy graph containing 8 pages is shown in Figure 2. Good pages are marked in white and bad ones in black. Bad pages 5, 6 and 8 make up a small link farm. Note that good page 1 points to page 5 and another good page 2 points page 1. Therefore if trust values are propagated via outlinks from page 1 or 2, it is difficult to distinguish bad ones like page 5.

Let us see what happens when computing without seeds by using the parameter \(\alpha = \beta = 0.5\), \(M = 50\), \(\alpha_d = 0.85\).

\[
\text{AVRank} = [0.07, 0.10, 0.18, 0.16, 0.21, 0.11, 0.07, 0.11]
\]

\[
\text{HVRank} = [0.08, 0.16, 0.18, 0.10, 0.20, 0.14, 0.07, 0.08]
\]

The result is disappointing. Page 5, 6 and 8 have very high AVRank scores. But if using a seed set, for example, we select page 2 and page 3 as good seeds and assign their initial AVRank and HVRank values to 0.5, we can get the following satisfactory results:

\[
\text{AVRank} = [0.07, 0.14, 0.35, 0.27, 0.08, 0.03, 0.02, 0.03]
\]

\[
\text{HVRank} = [0.02, 0.18, 0.31, 0.22, 0.07, 0.07, 0.10, 0.02]
\]

Page 5's AVRank decreases sharply, page 6 to page 8 have very low AVRanks. At the same time, the HVRank of page 1 decreases because it points to page 5. The HVRank of page 7 increases due to an outlink to page 4. Although it is just a toy graph, it indicates that our algorithm is effective while judging pages’ authority values and hub values. The result is more reasonable when an appropriate seed set is involved.

4. SEED SELECTION

As mentioned above, seed selection is of significant importance for these biased PageRank algorithms to combat web spam. In this section, we will mainly discuss how to select the seeds based on the web structure so as to maximize the seeds propagation coverage. The result bias towards the seed set is another focal point whom we pay attention to.

4.1 Propagation Coverage

The first problem we need to concern about a seed set is its propagation ability, i.e., what percentage of pages or sites can be covered from a seed set. If merely a few pages can be covered through a seed set (compared with the total number), its effect is limited even if its quality is very high.
In fact, the propagation coverage of a seed set is determined by web graph structure to some extent. From previous work [20, 2] and our experiments, the web graph can be shown as that in Figure 3, which includes IN, OUT, SCC, Tendrils, Tunnel and ISLANDs. The WCC (weakly connected component) consists of all of the parts, except ISLANDs.

In general, selecting reputable pages carefully as seeds to form a small seed set is a widely used method. As we know, most famous and reputable sites are in SCC (see experimental results in Section 5). So whatever method of propagation is used, the pages that can be covered from the seed set are nearly fated: all pages in SCC, OUT, and some pages in IN and other parts.

The CPV algorithm achieves a better result of propagation because both of directional link information of a page are used. No matter where the seeds locate (SCC, IN, or OUT), they can cover the same parts of the web graph: the whole SCC, OUT and Tunnel, most IN and Tendrils except the pages with no incoming links. From this respect, the CPV algorithm surpasses the single-direction algorithm with a stronger ability of propagation. Table 1 shows the comparison among different algorithms. Actually seeds located in SCC and some main ISLANDs can be selected to maximize the propagation coverage.

Table 1: Comparison of propagation coverage among different algorithms

<table>
<thead>
<tr>
<th>Position of Seed</th>
<th>Propagation</th>
<th>Coverage Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCC</td>
<td>Forward</td>
<td>SCC</td>
</tr>
<tr>
<td>IN</td>
<td>Forward</td>
<td>SCC, OUT, some IN</td>
</tr>
<tr>
<td>OUT</td>
<td>Forward</td>
<td>some SCQ</td>
</tr>
<tr>
<td>SCC</td>
<td>Backward</td>
<td>SCC, IN</td>
</tr>
<tr>
<td>IN</td>
<td>Backward</td>
<td>some IN</td>
</tr>
<tr>
<td>OUT</td>
<td>Backward</td>
<td>IN, SCC, some OUT</td>
</tr>
<tr>
<td>SCC/IN/OUT</td>
<td>Bidirectional</td>
<td>most WCC (with AVRank)</td>
</tr>
<tr>
<td>SCC/IN/OUT</td>
<td>Bidirectional</td>
<td>full WCC (AVRank &amp; HVRank)</td>
</tr>
</tbody>
</table>

Without the limitation of the location of the seeds, thus, seeds can be selected more easily. Differing from the carefully selected small seed set, a large number of seeds can be selected automatically. For example, only pages in the .gov and .edu domain are considered as seeds. No doubt that this process is time saving. So when using a large seed set, we can gain high propagation coverage as well as simplification of selecting process.

4.2 Result Bias

Using a small seed set has another big problem: result bias. That is, the top ranking results have a strong bias towards seeds.

The bias is due to the damping factor. From the PageRank formula, each seed will be refilled with \((1 - \alpha_d) \cdot 1/N_s\), where \(N_s\) is the number of seeds. It is a relatively large number when the number of seed set is small. So seeds can occupy most top positions in the final result. To reduce this result bias, a feasible way is to increase the number of seeds.

The sites appear in both of these two top-500 groups (i.e. the intersection of these two top-500 groups) are selected as seed candidates. There are 46 candidates found in our experiment, most (44 sites) of which are ranked in top 1000 in PageRank. It has a strong bias since 45 (97.8%) of them belong to SCC. Then we manually evaluate these candidates and eliminate those pure directories and non-good sites. The final seed set consists of 38 sites. This small seed set is named as seed set \(\mathcal{X}\) in the following parts.

Besides the small seed set \(\mathcal{X}\), a seed set with a large quantity of sites is also chosen. To keep the quality of the seeds at a relatively high level, those government sites (.gov or .gov.cn) and education sites (.edu or .edu.cn) from our data set are the first choice. These sites have both good authority values and hub values except few outliers. At last 9,003 education sites and 14,663 government sites are placed into the large seed set which is named as \(\mathcal{L}\) in the experiments.

Table 2: The Component Distribution of \(\mathcal{L}\)

<table>
<thead>
<tr>
<th>Position</th>
<th>Edu Sites</th>
<th>Gov Sites</th>
<th>Total (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCC</td>
<td>3112</td>
<td>6372</td>
<td>9484 (40.1%)</td>
</tr>
<tr>
<td>IN</td>
<td>2121</td>
<td>3280</td>
<td>5401 (22.8%)</td>
</tr>
<tr>
<td>OUT</td>
<td>1133</td>
<td>1812</td>
<td>2945 (12.4%)</td>
</tr>
<tr>
<td>TUNNEL</td>
<td>13</td>
<td>10</td>
<td>23 (1.0%)</td>
</tr>
<tr>
<td>T.OUT</td>
<td>179</td>
<td>175</td>
<td>354 (1.5%)</td>
</tr>
<tr>
<td>ISLAND</td>
<td>2322</td>
<td>2875</td>
<td>5197 (22.0%)</td>
</tr>
<tr>
<td>Total</td>
<td>9003</td>
<td>14663</td>
<td>23666 (100%)</td>
</tr>
</tbody>
</table>

The component distribution of \(\mathcal{L}\) is given in Table 2. These sites scatter in every part of the web graph. The size of each part in the seed set is approximately proportional to the whole graph. The amount and the diversity of \(\mathcal{L}\) provide us a better propagation coverage and less bias.

For deep analysis of the features of \(\mathcal{L}\), Figure 4 shows the PageRank distribution of this seed set \(\mathcal{L}\). The PageRank is still approximately following the power-law distribution. All of the sites have rank positions prior to the 10^5th and most of them have a high PageRank score. Therefore, these .gov and .edu sites are trustworthy ones and they are suitable to be seeds.

4.4 Bad Seeds

Besides using a good seed set to propagate trust, bad seeds can also be used for propagating distrust [19]. Spam sites are always linking to each other to form an alliance, and therefore a “powerful” bad seed set will be helpful for spam detection. However, selecting a bad seed set is more difficult than selecting a good one. In previous work [30], the authors illustrate how to select a bad seed set of “high values” effectively. They first get two ranking lists

\[\gamma = 10 \quad \text{and} \quad \alpha_d = 0.85, \text{if the 1000th page has a score of } 4 \times 10^{-5}, \text{we can get } N_f = 3750.\]
which contain top 1,000 most decreased sites each (compared with PageRank) in their ranking positions with different good seeds. Therefore, bad seeds can be manually selected from the intersection of these two ranking lists.

In our experiment, we will show that a bad seed set cannot help a large (good) seed set, though it does help a small one. Thus we do not need to manually select such a seed set at all.

5. EXPERIMENT

5.1 Data Set

To evaluate our CPV algorithm, we conducted experiments on a set of pages crawled by ourselves. It contains 13,285,711 pages from 358,245 hosts. Since the experiments were performed at site-level, this data set is slightly larger than that of WEBSPAM-UK2007 (which has 114,529 hosts). There are many tiny islands in our data set, however, this does not affect our analysis.

![Figure 4: The PageRank distribution of L](image)

<table>
<thead>
<tr>
<th>Part</th>
<th>Number of Sites (Percentage)</th>
<th>Number of Pages (Percentage)</th>
<th>PageRank Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCC</td>
<td>71023 (19.83%)</td>
<td>8875831 (66.81%)</td>
<td>75.28%</td>
</tr>
<tr>
<td>IN</td>
<td>99075 (27.66%)</td>
<td>2756758 (20.75%)</td>
<td>11.5%</td>
</tr>
<tr>
<td>OUT</td>
<td>39746 (11.09%)</td>
<td>563724 (4.24%)</td>
<td>7.96%</td>
</tr>
<tr>
<td>Tunnel</td>
<td>673 (0.19%)</td>
<td>9979 (0.08%)</td>
<td>0.05%</td>
</tr>
<tr>
<td>TIN</td>
<td>7963 (2.22%)</td>
<td>56931 (0.43%)</td>
<td>0.42%</td>
</tr>
<tr>
<td>T.OUT</td>
<td>12377 (3.45%)</td>
<td>136887 (1.03%)</td>
<td>0.53%</td>
</tr>
<tr>
<td>ISLAND</td>
<td>127388 (35.56%)</td>
<td>885601 (6.67%)</td>
<td>4.26%</td>
</tr>
</tbody>
</table>

Table 3: Web graph components

<table>
<thead>
<tr>
<th>Part</th>
<th>Number of Sites (Percentage)</th>
<th>Number of Pages (Percentage)</th>
<th>PageRank Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCC</td>
<td>71023 (19.83%)</td>
<td>8875831 (66.81%)</td>
<td>75.28%</td>
</tr>
<tr>
<td>IN</td>
<td>99075 (27.66%)</td>
<td>2756758 (20.75%)</td>
<td>11.5%</td>
</tr>
<tr>
<td>OUT</td>
<td>39746 (11.09%)</td>
<td>563724 (4.24%)</td>
<td>7.96%</td>
</tr>
<tr>
<td>Tunnel</td>
<td>673 (0.19%)</td>
<td>9979 (0.08%)</td>
<td>0.05%</td>
</tr>
<tr>
<td>TIN</td>
<td>7963 (2.22%)</td>
<td>56931 (0.43%)</td>
<td>0.42%</td>
</tr>
<tr>
<td>T.OUT</td>
<td>12377 (3.45%)</td>
<td>136887 (1.03%)</td>
<td>0.53%</td>
</tr>
<tr>
<td>ISLAND</td>
<td>127388 (35.56%)</td>
<td>885601 (6.67%)</td>
<td>4.26%</td>
</tr>
</tbody>
</table>

Table 4: Distribution of top-ranked sites by PageRank

<table>
<thead>
<tr>
<th>Part</th>
<th>SCC</th>
<th>OUT</th>
<th>IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>top 10</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>top 100</td>
<td>88%</td>
<td>12%</td>
<td>0%</td>
</tr>
<tr>
<td>top 1000</td>
<td>82.4%</td>
<td>16.3%</td>
<td>0.2%</td>
</tr>
<tr>
<td>top 10000</td>
<td>77.93%</td>
<td>20.37%</td>
<td>0.52%</td>
</tr>
</tbody>
</table>

In order to obtain the web structure graph from our data set, Kosaraju Algorithm [8] is used to find the SCC (strongly connected component) and then figure out the remainder structure. Kosaraju Algorithm finishes in time $O(n + e)$ where $n$ is the number of vertices and $e$ is the number of edges. Since the web graph is sparse, the running time is acceptable even when the number of vertex reaches billion. Our computation is based on site-level. The result is listed in Table 3. There are nearly 60% sites and 90% pages in SCC, IN and OUT. They approximately possess all of the PageRank scores. Although the percentage of island sites is a little bit high (35%), fortunately there are only a few of pages (6.67%) in these independent sites. The further exploration of the distribution of top-ranked sites is shown in Table 4. Obviously, most of top-ranked sites are located in SCC and OUT.

5.2 Parameter Selection

As used in many other experiments, the parameter $\alpha_q$ is set to 0.85. Different parameters are tested ($\alpha$ and $\beta$) out of the following two considerations:

- By symmetry, $\beta = 1 - \alpha$, which means if a page’s AVRank is determined by its parent pages’ AVRanks at percentage of $\alpha$, then its HVRank is determined by its child pages’ AVRanks at percentage of $1 - \alpha$.
- By visit model. To avoid a too complicated visit model, we simply keep $\beta = 1$. From Eq.(A.1), this means a user has a probability of $\alpha$ to do a random surfer and a probability of $1 - \alpha$ to back once and surfer (when not using a damping factor).

We use Kendall’s-$\tau$ correlation to measure the similarity between AVRank/HVRank and PageRank. Table 5 shows the result when $\alpha$ varies from 0 to 1. AVRank is very close to PageRank which indicates that it is a good metric to assess pages’ authority values. HVRank is less correlative with PageRank only because it is a metric of pages’ hub values. There is no much difference between different parameters. In the following experiments, $\alpha = \beta = 0.5$ is used because this set of parameter not only embodies symmetry, but also has a sound visit mode - when not using a damping factor, a user does random surfer at the probability of 50%, backs once and continues random surfer at the probability of 25%, backs twice or more and continues random surfer at the probability of 25%.

AVRank or HVRank scores are often used solely to measure a page’s importance. Obviously they can work together. We have the following four prevalent ways of mean:

- arithmetic mean
- geometric mean
- harmonic mean
- quadratic mean (root mean square)

The Kendall’s-$\tau$ correlation coefficient of each mean and PageRank are all considered. In this dataset, the arithmetic mean whose Kendall’s-$\tau$ correlation is 0.95 is very close to PageRank. The other three are all about 0.50, which is a little far from PageRank. But the result of quadratic mean is better than the other three mean mixtures when checking the top results manually. So quadratic mean is used as our combination method of AVRank and HVRank in the following experiments.

5.3 The Propagation Coverage of Two Good Seed Sets

In the section 4, two seed sets which have different size are mentioned. One is a small seed set $X$ and the other is a large seed set $L$. In order to make clear the pros and cons of these two seed sets, we compare the propagation capability of these two seed sets in different algorithms. The propagation coverage is employed to evaluate the propagation capability of a seed set. It is the percentage of the pages that can be reached via links after a number of iterations.

The propagation coverage of TrustRank and CPV algorithm with the seed set $X$ is given in Table 6. $Sn(s)$ denotes the number of sites that can be covered from seed set $s$. If we use only forward
Table 5: Kendall’s-τ correlation between AVRank/HVRank and PageRank

<table>
<thead>
<tr>
<th>α</th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ</td>
<td>0.94</td>
<td>0.95</td>
<td>0.95</td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>τ</td>
<td>0.94</td>
<td>0.95</td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>τ</td>
<td>0.21</td>
<td>0.22</td>
<td>0.23</td>
<td>0.23</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>τ</td>
<td>0.21</td>
<td>0.22</td>
<td>0.23</td>
<td>0.23</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

propagation like TrustRank, it can only cover the whole SCC and OUT. The coverage of TrustRank in our dataset can prove this well. It can cover one third of the sites. With CPV algorithm, AVRank covers most WCC except those which do not have incoming links. Quadratic Mean can cover the whole WCC and the percentage is twice over TrustRank.

Table 6: Propagation coverage with \( X \)

<table>
<thead>
<tr>
<th>Sn(( X )) (Percentage)</th>
<th>TrustRank</th>
<th>110769 (30.92%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVRank</td>
<td>132427 (36.97%)</td>
<td></td>
</tr>
<tr>
<td>HVRank</td>
<td>193986 (54.15%)</td>
<td></td>
</tr>
<tr>
<td>Quadratic Mean</td>
<td>230857 (64.44%)</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Propagation coverage with \( L \)

<table>
<thead>
<tr>
<th>Sn(( L )) (Percentage)</th>
<th>TrustRank</th>
<th>123383 (34.44%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVRank</td>
<td>142605 (39.81%)</td>
<td></td>
</tr>
<tr>
<td>HVRank</td>
<td>202078 (56.41%)</td>
<td></td>
</tr>
<tr>
<td>Quadratic Mean</td>
<td>238015 (66.44%)</td>
<td></td>
</tr>
</tbody>
</table>

5.4 The Quantity of Seeds vs. The Result Bias

We have mentioned that the small seed set has a strong bias to the seeds in the ranking results in section 4. Increasing the quantity of the seed set can overcome this defect to some extent. For the purpose of figuring out the relationship between the quantity of seeds and the result bias, we deepen our experiment by using different size of the seed set.

We start from 50 seeds and double the number each time. At each point, we randomly select seeds 4 times and calculate the average number of seeds that top 100 and top 1000 results contain.

Table 8: Result Bias

<table>
<thead>
<tr>
<th>number of seeds</th>
<th>number of seeds in top 100 results</th>
<th>number of seeds in top 1000 results</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>100</td>
<td>95.5</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>92</td>
<td>200</td>
</tr>
<tr>
<td>400</td>
<td>78.75</td>
<td>400</td>
</tr>
<tr>
<td>800</td>
<td>49.25</td>
<td>800</td>
</tr>
<tr>
<td>1600</td>
<td>21.75</td>
<td>812</td>
</tr>
<tr>
<td>3200</td>
<td>15</td>
<td>587.25</td>
</tr>
<tr>
<td>6400</td>
<td>18.25</td>
<td>419.25</td>
</tr>
<tr>
<td>12800</td>
<td>33.5</td>
<td>428.5</td>
</tr>
</tbody>
</table>

The outcome of the same experiments made with the large seed set \( L \) is shown in Table 7. It lists the propagation coverage of TrustRank and our algorithm. Similarly, CPV algorithm has a better propagation coverage than TrustRank does. Quadratic Mean can reach nearly two thirds of the total sites.

Compared with the propagation coverage using the small seed set \( X \), the improvement is not so remarkable. The propagation coverage of all the algorithms is just a slight increase. This result disagrees on our intuitive expectation and the reason is there are so many tiny islands in our data set, which inevitably limit the propagation of seeds.

Figure 5: Average number of seeds in top 100 results

5.4 The Quantity of Seeds vs. The Result Bias

We have mentioned that the small seed set has a strong bias to the seeds in the ranking results in section 4. Increasing the quantity of the seed set can overcome this defect to some extent. For the purpose of figuring out the relationship between the quantity of seeds and the result bias, we deepen our experiment by using different size of the seed set.

We start from 50 seeds and double the number each time. At each point, we randomly select seeds 4 times and calculate the average number of seeds that top 100 and top 1000 results contain. Table 8 shows the result. When the seed set is small, the top results are nearly all occupied by seeds. The number of seeds in top 100 results reaches the lowest at the case of 3000 seeds and the ratio runs into stable at about 4.000. In order to prove our assumption in Section 4.2, we check the scores of sites in about 1000th position. The 1000th site’s AVRank score is about 3.98 \times 10^{-5}, which is well matched with our estimation.
5.5 Anti-Spamming

In order to compare the ability of anti-spamming, a set of sample sites with known types is needed. A sample of 1000 sites which can give enough information and still be manageable is settled. Using a method similar to that in TrustRank [15], a list of sites in descending order of their PageRank scores is generated and segmented into 20 buckets. Each of the buckets contains a different number of sites with scores summing up to 5 percent of the total PageRank score. We construct a sample set of 1000 sites by selecting 50 sites at random from each bucket. Then we perform a manual evaluation to determine their categories. Each site is classified into one of the following categories: reputable, spam, pure directory, and personal blog. A sample site who uses any spamming techniques is put in the spam category. Not available sites are thrown away and other substitution ones are reselected in the sampling process. The distribution of our sample is presented in Figure 9.

5.5.1 Without Seeds

We first take a look at the anti-spamming ability of our method without seed sets. Figure 7 shows the ratio of reputable, directory and personal blog sites in each bucket. The buckets are divided the same as our sampling phase. The horizontal axis represents the bucket numbers. The vertical axis corresponds to the percentage of each sample part. The reputable and directory sites are considered as good ones. Their contributions are shown in white and middle gray segments. The personal blogs are marked in dark gray segment because the quality of them cannot be simply judged. Spam sites are not drawn so that the height of bars can represent the ratio of good sites. We pay more attention to the first 10 buckets because these 10 buckets contain the top 50% results.

Figure 7 shows that first 3 buckets have a high ratio of good sites, but there are so many personal blogs in these buckets. There are spam sites in every buckets, especially the results from bucket 4 to bucket 10 are not so satisfied and have plenty of room to be improved. We can also find that our algorithm has very little promotion without a seed set.

5.5.2 With a Small Seed Set

As a comparison of the effect of anti-spamming with the seed set $\mathcal{S}$, We experimentally show the TrustRank and CPV algorithm using seed set $\mathcal{X}$. Figure 8 illustrates the distributions of spam site in TrustRank and CPV buckets. The very first 2 buckets have the excellent results because of result bias. The seeds and the sites that are directly pointed by seeds take most top ranking positions which causes the number of sample sites in first 2 buckets is small. In the following 8 buckets, however, the results of TrustRank and AVRank reveal the true nature. TrustRank has spam sites no more than 5 in top 10 buckets. AVRank has a better result in top 8 buckets. The number of spam sites in bucket 9 is nearly 10 because of the demotion of spam sites in the first 9 buckets. The Quadratic Mean has a balanced and fairly good result in first 9 buckets, the average number of spam sites in these buckets is only 1.7. But the spam sites increase sharply from bucket 10, which shares the same reason as AVRank.

5.5.3 With a Large Seed Set

The results of anti-spamming using seed set $\mathcal{L}$ are displayed in this section. From Figure 10, we can see that:

(1) Compared with small seed set $\mathcal{X}$, there are remarkable im-

![Figure 7: Good sites in Page-Rank, AV-Rank and Quadratic Mean buckets (with no seeds)](image)

![Figure 8: Bad sites in Trust-Rank, AV-Rank and Quadratic Mean buckets (with $\mathcal{X}$)](image)

![Figure 9: Distribution of categories in the evaluation sample](image)
From the analysis above, we can draw conclusion that a large seed set can perform better than a small seed set does in detecting spam sites. The main reason is that our algorithm with a large seed set gives the result of TrustRank and AVRank with the seed set \( \mathcal{X} \). The demotions of our algorithm and TrustRank are at the same level in general. Most reputable sites still have high rankings.

In Figure 12 and Figure 13, good sites (reputable and directory) are not top-ranked, whereas spam sites are demoted to the last 5 buckets. The diversity of large seed set \( \mathcal{L} \) makes the ranking position of these samples change a lot.

(2) CPV algorithm is better than TrustRank in most of the first 10 buckets. The Quadratic Mean especially has an excellent result with 100% in the first 6 buckets. The buckets are nearly occupied by good sites and the ratio of the white segment increases. Most of spam sites are demoted to the last 5 buckets. The black segment in Figure 11 is larger than in Figure 12. The bad seed set consists of 43 bad sites which are selected with a method same to [30]. The initial values of these bad seeds are summed up to \(-1\). They are equally divided by the number of seeds.

The results of combination of two seed sets is shown in Figure 11. Unfortunately, the result has no improvement in combating spam. The main reason is that our algorithm with a large seed set can do the job well enough. In this scenario, it does not help when using a bad seed set, especially a small one. However, we believe a large bad seed set of “high quality” is undoubtedly useful for combating spam.

5.5.4 With a Mixed Seed Set

In this part, a bad seed set \( \mathcal{B} \) is combined with a large seed set \( \mathcal{L} \). The bad seed set consists of 43 bad sites which are selected with a method same to [30]. The initial values of these bad seeds are summed up to \(-1\). They are equally divided by the number of seeds.

The results of combination of two seed sets is shown in Figure 11. Unfortunately, the result has no improvement in combating spam. The main reason is that our algorithm with a large seed set can do the job well enough. In this scenario, it does not help when using a bad seed set, especially a small one. However, we believe a large bad seed set of “high quality” is undoubtedly useful for combating spam.

5.5.5 The Demotion of Spam Sites and Non-spam Sites

We have already known that our method can better demote the spam sites whenever using seed set \( \mathcal{X} \) and \( \mathcal{L} \). The relative stableness of non-spam sites is still the key point of a ranking algorithm. So we check the demotion of non-spam sites in this part. Figure 12 gives the result of TrustRank and AVRank with the seed set \( \mathcal{X} \). The demotions of our algorithm and TrustRank are at the same level in general. Most reputable sites still have high rankings.
an obvious division at value 4 between spam sites and non-spam sites. The large seed set $\mathcal{L}$ performs much better than using $\mathcal{X}$. The demotion of spam sites are all above 4 when using $\mathcal{L}$.

![Figure 14: Bucket-level demotion of AV-Rank scores](image)

6. CONCLUSION

In this paper, we propose two new metrics, AVRank and HVRank, to measure a page’s value. By exploiting the bidirectional links, these two metrics can work more effectively. We also analyze the shortcoming of small seed sets and discuss the result bias of seeds. The experimental results show that using a large seed set can achieve a better performance than using a small one. In addition, choosing a large number of seeds automatically is actually easier than choosing a small number of seeds manually and carefully. The experimental results also show that our proposed CPV algorithm performs better than TrustRank. Although AVRank and HVRank can work independently, the quadratic mean of them exhibits the best anti-spamming ability.

There still leaves some unanswered questions in our research. For example, ranking algorithms used to anti-spamming try to decrease the scores and positions of spam pages but they cannot directly identify the spam pages effectively. We will study such problems in our future work.

7. REFERENCES

[29] H. Yang, I. King, and M. R. Lyu. Diffusionrank: a possible


APPENDIX

A. PROOF OF THE CONVERGENCE

According to the matrix theory [9, 17], we have the following lemmas:

**Lemma 1.** Column-stochastic matrix $A$ has dominant eigenvalue $\lambda_1 = 1$, $1 \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0$.

**Lemma 2.** Strictly diagonally dominant matrix is invertible.

**Lemma 3.** If the series $f(\lambda) = \sum_{i=0}^{\infty} a_i \lambda^i$ is convergent for all values of $\lambda$ such that $|\lambda| < R$, the matrix series $F(A) = \sum_{i=0}^{\infty} a_i A^i$ is also convergent if $A$ has $\lambda$ as an eigenvalue.

Now, we start to prove the theorem below:

**Theorem 1.** The computation of $\text{AVRank}$ and $\text{HVRank}$ is convergent.

**Proof.** In section 3, we give the formula of the $\text{AVRank}$ and $\text{HVRank}$:

$$\text{AVRank} = \alpha M^T \cdot \text{AVRank} + (1 - \alpha)M^T \cdot \text{HVRank}$$

$$\text{HVRank} = \beta N^T \cdot \text{AVRank} + (1 - \beta)N^T \cdot \text{HVRank}.$$ 

We expand $\text{AVRank}$ by using $\text{HVRank}$:

$$\text{AVRank} = \alpha M^T \cdot \text{AVRank} + (1 - \alpha)M^T \cdot (\beta N^T \cdot \text{AVRank} + (1 - \beta)N^T \cdot \text{HVRank})$$

$$= \alpha M^T \cdot \text{AVRank} + (1 - \alpha)\beta M^T \cdot N^T \cdot \text{AVRank} +$$

$$(1 - \alpha)(1 - \beta)M^T \cdot N^T \cdot \text{HVRank}$$

$$\alpha M^T \cdot \text{AVRank} + (1 - \alpha)\beta M^T \cdot N^T \cdot \text{AVRank} +$$

$$(1 - \alpha)(1 - \beta)M^T \cdot (N^T)^2 \cdot \text{AVRank} + \cdots$$

$$= \alpha M^T + (1 - \alpha)\beta M^T \cdot N^T +$$

$$(1 - \alpha)(1 - \beta)M^T \cdot (N^T)^2 + \cdots) \cdot \text{AVRank}$$

(A.3)

$$\text{AVRank} = [\alpha M^T + (1 - \alpha)\beta M^T \cdot N^T +$$

$$(1 - \alpha)(1 - \beta)M^T \cdot (N^T)^2 + \cdots)] \cdot \text{AVRank}$$

(A.4)

Since $N^T$ is a column-stochastic matrix, its power $(N^T)^j(i \geq 1)$ is also a column-stochastic matrix. According to Lemma 1, $(N^T)^j$ has dominant eigenvalue $\lambda_1 = 1$. Given that $0 \leq \beta \leq 1$ and unit matrix $I$ also has an eigenvalue $\lambda = 1$, if $\beta > 0$, the series

$$f(\lambda) = \sum_{i=0}^{\infty} (1 - \beta)^i \lambda^i = \sum_{i=0}^{\infty} (1 - \beta)^i$$

is convergent. By Lemma 3, the matrix series

$$\sum_{i=0}^{\infty} (1 - \beta)^i (N^T)^j$$

is convergent, so the matrix series

$$(1 - \alpha)\beta M^T \cdot N^T \cdot \sum_{i=0}^{\infty} (1 - \beta)^i (N^T)^j$$

is also convergent.

On the other hand,

$$\left[\sum_{i=0}^{\infty} (1 - \beta)^i (N^T)^j \right] \cdot (1 - (1 - \beta)N^T)$$

$$= \sum_{i=0}^{\infty} (1 - \beta)^i (N^T)^j - \sum_{i=0}^{\infty} (1 - \beta)^i (N^T)^j$$

$$= 1$$

Since $I - (1 - \beta)N^T$ is a strictly diagonally dominant matrix, according to Lemma 2, it is invertible. Thus, we can have the following equation:

$$(1 - \alpha)\beta M^T \cdot N^T \cdot \left(\sum_{i=0}^{\infty} (1 - \beta)^i (N^T)^j \right)$$

$$= (1 - \alpha)\beta M^T \cdot N^T \cdot (1 - (1 - \beta)N^T)^{-1}$$

$I - (1 - \beta)N^T$ is a matrix in which every column has the same summation $1 - (1 - \beta) = \beta$. Clearly its inverse matrix’s each column also has the same summation $1/\beta$. So

$$(1 - \alpha)\beta M^T \cdot N^T \cdot (I - (1 - \beta)N^T)^{-1}$$

is a matrix in which each column has the same summation $(1 - \alpha)\beta \cdot (1/\beta) = 1 - \alpha$. As we know that $\alpha M^T$ is a matrix whose every column-summation is $\alpha$, we now have a column-stochastic matrix:

$$\alpha M^T + (1 - \alpha)\beta M^T \cdot N^T \cdot (I - (1 - \beta)N^T)^{-1}$$

Now that $\text{AVRank}$ is the dominant eigenvector under eigenvalue $\lambda_1 = 1$, we can use power iteration to compute it and the result is convergent.

If $\beta = 0$, Eq. (A.4) is simplified as $\text{AVRank} = \alpha M^T \cdot \text{AVRank}$. Since $\alpha M^T$ has dominant eigenvalue $\lambda_1 = \alpha$, $\text{AVRank}$ is exactly the dominant eigenvector under eigenvalue $\alpha$.

The case of $\text{HVRank}$ is basically same as that of $\text{AVRank}$. We omit the proof due to the page limitation. □