DirichletRank: Solving the Zero-One Gap Problem of PageRank

XUANHUI WANG and TAO TAO
University of Illinois at Urbana-Champaign
JIAN-TAO SUN
Microsoft Research Asia
and
AZADEH SHAKERY and CHENGXIANG ZHAI
University of Illinois at Urbana-Champaign

Link-based ranking algorithms are among the most important techniques to improve web search. In particular, the PageRank algorithm has been successfully used in Google search engine and is attracting much attention recently. However, we find that PageRank has a "zero-one gap" problem which, to the best of our knowledge, has not been addressed in any previous work. This problem can be potentially exploited to spam PageRank results and make the state-of-the-art link-based anti-spamming techniques ineffective. The "zero-one gap" problem arises as a result of the current ad hoc way of computing the transition probabilities in the random surfing model. We therefore propose a novel *DirichletRank* algorithm which calculates these probabilities using Bayesian estimation with a Dirichlet prior. DirichletRank is a variant of PageRank, but it does not have the problem of "zero-one gap" and can be analytically shown to be substantially more resistant to some link spams than PageRank. Experiment results on TREC data show that DirichletRank can achieve better retrieval accuracy than PageRank due to its more reasonable allocation of transition probabilities. More importantly, experiments on the TREC data set and another real web data set from WebGraph project show that, compared with the original PageRank, DirichletRank is more stable under link perturbation and is significantly more robust against both manually identified web spams and several simulated link spams. DirichletRank can be computed as efficiently as PageRank, and thus it is scalable to large-scale web applications.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*

General Terms: Algorithms

Additional Key Words and Phrases: DirichletRank, PageRank, zero-one gap, spamming, link analysis

Author's address: X. Wang, A. Shakery, and C. Zhai, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. Email: {xwang20, shakery, czhai}@cs.uiuc.edu. T. Tao, Microsoft Corporation, Redmond, WA 98052. Email: taotao@microsoft.com. J.-T. Sun, Microsoft Research Asia, Beijing, China 100080. Email: jtsun@microsoft.com.

1. INTRODUCTION

Web search is becoming a dominant method of information acquisition in our daily life because of the explosive growth of online resources. Link-based ranking algorithms have proven to be key to the success of web search engines. PageRank [Brin and Page 1998; Page et al. 1999] and HITS [Kleinberg 1999] are the two earliest link analysis algorithms to identify "authoritative" pages via hyperlink information. Since then, many methods have been developed to improve the accuracy of these algorithms (*e.g.*, [Bharat and Henzinger 1998; Haveliwala 2002; Li et al. 2002]).

While the PageRank algorithm has been successfully used in Google search engine and is attracting a lot of research attention recently, it has a "zero-one gap" problem, which arises from the ad hoc way of computing the transition probabilities in the random surfing model adopted in the existing work. In PageRank, a fictious random web surfer visits the web pages one by one continuously. At each page, the surfer would choose the next one by either following the out-links of the current page with probability $1 - \lambda$ or jumping to a random page with probability λ. To our best knowledge, all existing variants of PageRank fix the value of λ to be a constant (often 0.15 [Brin and Page 1998]) for every page except "dangling pages" (pages without out-links), whose λ's are set to be 1 [Bianchini et al. 2005]. As a result, the λ values of a page with no out-link and a page with a single out-link have an unreasonably large gap. We refer to this problem as "zero-one gap". On the surface, we may reduce the gap by increasing λ, but a large λ value cannot effectively exploit hyperlink information. Thus the "zero-one gap" problem is inherent in PageRank.

The "zero-one gap" is indeed a flaw, since it can be potentially exploited to manipulate PageRank results by *link spamming* [Gyongyi and Garcia-Molina 2005a]: Spammers can intentionally set up many interconnected pages to boost the PageRank scores of a small number of target pages. For instance, Figure 1 illustrates a typical link spam structure, where we use *leakage* to denote the PageRank scores propagated to the link farm from external pages. In this structure, a web site owner creates a large number of bogus web pages B's (*i.e.*, pages whose sole purpose is to promote the target page's ranking score), all pointing to and pointed by a single target page T. As will be detailed in the rest of this paper, the "zero-one gap" problem makes PageRank sensitive to this type of structures and tend to assign a much higher ranking score to T than it deserves (up to 10 times easily).

Furthermore, the "zero-one gap" problem makes PageRank sensitive to certain local structure changes, and thus *unstable*, which could reduce the effectiveness of the state-of-the-art link-based anti-spamming techniques since they are all based on the original PageRank algorithm. For example, the TrustRank [Gyongyi et al. 2004] and its variant Topical TrustRank [Wu et al. 2006] initialize trust scores from well-known and trustworthy seed sites, and propagate them through hyperlinks using PageRank-like methods. Since TrustRank can also be *leaked* to a spam page, this spam page can manipulate its TrustRank score easily by adding a few bogus pages similarly in Figure 1.

To our best knowledge, this "zero-one gap" problem has not been discovered or addressed in any existing work. To solve it, we propose a novel *DirichletRank* algorithm which, *with comparable computational cost* to PageRank, calculates the

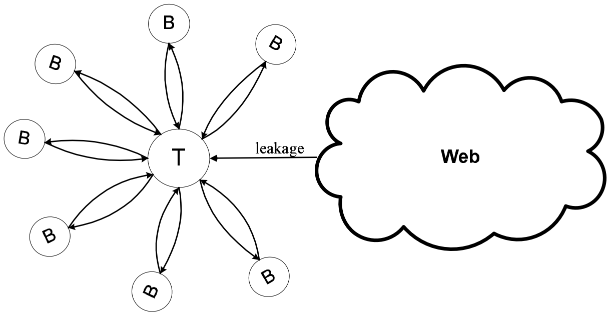


Fig. 1. An example of the link spams using bogus pages

transition probabilities using Bayesian estimation with a Dirichlet prior. DirichletRank is a variant of PageRank, but it does not have the problem of "zero-one gap" since in DirichletRank, the probability of jumping from a page to a random page is a smooth function of the current page's out-degree (see Equation (6)). Furthermore, DirichletRank can be analytically shown to be substantially more resistant to typical link farm spams such as Figure 1 than PageRank.

Experiment results on TREC data show that DirichletRank can achieve better retrieval accuracy than PageRank due to its more reasonable allocation of transition probabilities. More importantly, experiments on the TREC data set and another real web data set from WebGraph project show that, compared with the original PageRank, DirichletRank is more stable under link perturbation and is significantly more robust against both manually identified web spams and several simulated link spams. DirichletRank can be computed as efficiently as PageRank, and thus it is scalable to large-scale web applications.

The rest of the paper is organized as follows. We first review the PageRank algorithm and discuss the problem of "zero-one gap" in Section 2. In Section 3, we present the proposed DirichletRank algorithm and analytically show its robustness in dealing with link spamming. Our experiment results are presented in Section 4 and we discuss related work in Section 5. Finally we conclude our work in Section 6. All proofs are given in the appendix.

2. THE "ZERO-ONE GAP" PROBLEM IN PAGERANK

In this section, we analyze PageRank's "zero-one gap" problem and show its vulnerability to link spamming.

2.1 Basic PageRank

The basic PageRank algorithm [Page et al. 1999; Brin and Page 1998] models the whole web as a directed graph $G(V, E)$ with a vertex set V of N pages and a directed edge set E. By collapsing multiple links between the same pair of pages, we represent this graph by an $N \times N$ binary-value adjacency matrix:

$$A = [a_{ij}]_{N\times N} \text{ where } a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

PageRank calculates the importance value of each page iteratively through the importance values of its in-link pages. Formally, for each page v, let N_v be the out-links of v, B_v be the in-links of v, and $r(v)$ be the PageRank score of v, respectively:

$$r(v) = \sum_{u \in B_v} \frac{r(u)}{|N_u|}.$$

Assume M is the row normalized matrix of A, the updating of PageRank scores can be expressed as:

$$\mathbf{r} = M^T \mathbf{r}.$$

When there is at least one non-zero entry in each row, mathematically the iterative updating will converge to M's principal eigenvector. However, the convergence is guaranteed only if M is irreducible and aperiodic [Motwani and Raghavan 1995]. In web applications, the latter is guaranteed but the former is not and may result in the "rank sink" problem. To avoid this problem, PageRank introduces a uniform matrix U ($U_{ij} = 1/N$) and interpolates it with the original matrix M with a damping factor $1 - \lambda$:

$$\tilde{M} = (1 - \lambda) \cdot M + \lambda \cdot U \tag{1}$$

where λ ($0 \leq \lambda \leq 1$) is the *random jumping* probability. The improved PageRank scores are calculated as:

$$\mathbf{r} = \tilde{M}^T \mathbf{r} = (1 - \lambda) M^T \mathbf{r} + \frac{\lambda}{N} \mathbf{e}_N$$

where $\mathbf{e}_N = (1, ..., 1)^T$ is a column vector consisting of N elements of 1, and λ is typically set to 0.15 [Brin and Page 1998]. The intuition behind the interpolation can be explained by a random surfing model. A surfer follows the out-links of a page with probability $1 - \lambda$ and uniformly jumps to random pages with probability λ. A page's PageRank score can be interpreted as the average probability that a surfer would visit this page after surfing the whole web for a sufficiently long time.

2.2 Solving the "Zero Out-Link" Problem

The basic PageRank assumes each row of the matrix M has at least one nonzero entry, *i.e.* the corresponding vertex in G has at least one out-link. Unfortunately, this assumption never holds in reality. Many web pages simply have no out-link at all. Moreover, it is very difficult to crawl the whole web and thus many web applications can only consider a subgraph. In these cases, even if a page has out-links, these out-links may have been removed when the whole web is projected to a subgraph. For example, in the WT10G[1] data, only $1,295,841$ out of $1,692,096$, roughly 77%, documents have out-links. Simply removing all the pages without out-links is not a solution because it generates new zero-out-link pages. Indeed, this "dangling page" problem has been identified in [Ding et al. 2002; Bianchini et al. 2005; Brin and Page 1998; Page et al. 1999; Eiron et al. 2004]. In [Bianchini et al. 2005], the authors analyzed previous solutions and show that they all boil

[1] http://es.csiro.au/TRECWeb/wt10g.html

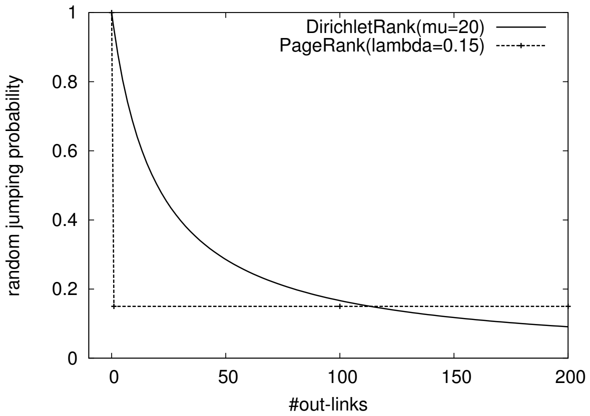


Fig. 2. The random jumping probability w.r.t the number of out-links

down to the following approach:

$$\tilde{M} = (1 - \lambda) \cdot M + \tilde{\lambda} \cdot U \tag{2}$$

where

$$\tilde{\lambda}(i) = \begin{cases} \lambda & \text{if } \sum_j M_{ij} = 1, \\ 1 & \text{otherwise.} \end{cases}$$

$\tilde{M}$ is a Markov matrix and guarantees the equilibrium distribution [Ding et al. 2002]. PageRank scores are:

$$\mathbf{r} = (1 - \lambda) M^T \mathbf{r} + \frac{\tau}{N} \mathbf{e}_N$$

where τ is the weighted sum of the random jumping probabilities,

$$\tau = \sum_{i=1}^{N} r(i) \times \tilde{\lambda}(i). \tag{3}$$

It is easy to see that $\tau = \lambda$ when $\tilde{\lambda}(i) = \lambda$ for $1 \leq i \leq N$.

As proved in [Bianchini et al. 2005], Equations (1) and (2) are equivalent in terms of final page ranking results, even if $\tilde{M}$ in Equation (1) is not a Markov matrix. In the rest of the paper, we use Equation (2) as the formula for PageRank.

2.3 The “Zero-One Gap” Problem

Although Equation (2) solved the “zero out-links” problem, there is another problem, which has not been addressed: The probability of jumping to random pages is 1 in a zero-out-link page, but it drops to λ (in most cases, $\lambda = 0.15$) for a page with a single out-link. We illustrate this problem in Figure 2. The dashed line in Figure 2 illustrates the random jumping probability of the PageRank algorithm with respect to the number of out-links. Clearly there is a big “gap” between 0 and 1 out-link. We refer to this problem as “zero-one gap”. (Note that the solid line in Figure 2 is DirichletRank to be discussed later.) On the surface, we can reduce

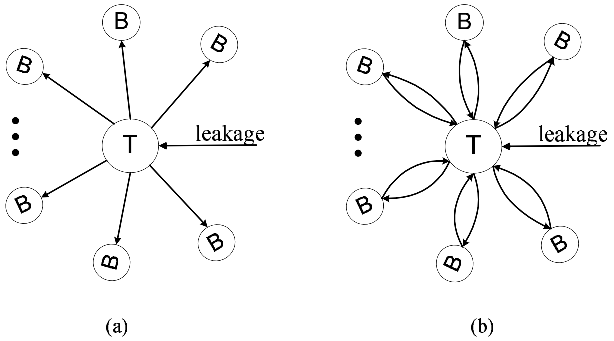


Fig. 3. Two contrast structures. The only difference between (a) and (b) is that all B's have back links to the target page T in (b).

the gap by increasing λ. Unfortunately, a large λ would give the algorithm little room to exploit the link information. Thus, the zero-one gap problem is inherent in PageRank.

The zero-one gap problem represents a flaw in PageRank as it allows a spammer to easily manipulate PageRank to achieve spamming. Consider the two structures in Figure 3: (a) is a case without link spamming; (b) represents a typical spamming structure with all bogus pages B's having back links to the target page T. (Here *leakage* denotes the PageRank scores propagated to the structures from external pages too.) In general, the number of bogus pages k satisfies $k \ll N$, and thus we can assume that τ in Figure 3 (a) equals to that in (b). Let $r_o(v)$ and $r_s(v)$ be the PageRank scores of any page v in Figure 3 (a) and (b) respectively and we have the following theorem.

THEOREM 1. *With k ($k \ll N$) bogus pages, σ leakage, and τ as the weighted sum of the random jumping probabilities defined in Equation (3),*

$$r_o(T) = \sigma + \frac{\tau}{N} \tag{4}$$

$$r_s(T) = \frac{1}{2\lambda - \lambda^2}\left[\sigma + \frac{\tau(k(1-\lambda)+1)}{N}\right], \tag{5}$$

and $r_s(T) \geq \frac{1}{2\lambda-\lambda^2} r_o(T)$ for any positive integer k.

The proofs of Theorem 1 and all the following theorems are given in the appendix.

Theorem 1 shows that $r_s(T)$ is larger than or equal to $r_o(T)$ for any positive integer k: Given $\lambda \in [0,1]$, $\frac{\partial}{\partial\lambda}\frac{1}{2\lambda-\lambda^2} = \frac{-2(1-\lambda)}{(2\lambda-\lambda^2)^2} \leq 0$. Thus, when $\lambda = 1$, $\frac{1}{2\lambda-\lambda^2}$ reaches its minimum value 1, and $r_s(T) \geq r_o(T)$ over the range of all λ values. On the other hand, $\lim_{\lambda\to 0}\frac{1}{2\lambda-\lambda^2} = \infty$. This means a small λ, usually preferred in PageRank [Page et al. 1999], can result in $r_s(T)$ much larger than $r_o(T)$. For example, when $\lambda = 0.15$, $r_s(T)$ is at least 3 times larger than $r_o(T)$. Note that the above analysis applies to all k's. Clearly, as k increases, their difference will be even more aggravated.

We see immediately an intrinsic problem in PageRank: when $k = 1$, there is only one bogus page in Figure 3 (b), but the addition of the back link of this bogus page makes the PageRank score of the target page 3 times larger than before. This is because a surfer is forced to jump back to the target page with a high probability in Figure 3 (b)[2]. Indeed, given the default value $\lambda = 0.15$, the single out-link in a bogus page forces a surfer to jump back to the target page with a probability 0.85!

The analysis above demonstrates that the "zero-one gap" represents a notable problem of PageRank, which makes it sensitive to a local structure change and thus vulnerable to link spamming. The "zero-one gap" is fundamental for all the PageRank-like methods which use a fixed λ to do the interpolation. For example, the state-of-the-art link-based anti-spamming algorithms TrustRank [Gyongyi et al. 2004] and Topical TrustRank [Wu et al. 2006] could be ineffective since spammers can manipulate their trust scores easily by simply building link structures similar to Figure 3 (b), once they can obtain some TrustRank *leakage* from other pages. In the following sections, we focus our discussion on solving the "zero-one gap" problem of the standard PageRank, but all the results are applied to all PageRank-like methods equally.

3. DIRICHLETRANK ALGORITHM

In this section, we derive a new algorithm called DirichletRank based on Bayesian estimation of transition probabilities. DirichletRank not only solves the problem of zero-one gap, but also provides a more principled way to solve the original zero-out-link problem. We analytically compare DirichletRank with PageRank and show that DirichletRank is less sensitive to local structure changes and more robust than PageRank.

3.1 Bayesian Estimation of Transition Probabilities

We note that the zero-one gap problem is caused by an unreasonable allocation of probabilities in the random surfing model. A natural solution is thus to seek for a more principled way to set such probabilities.

Let us assume that, in our random surfing model, the probabilities of a surfer transiting from a page v to other pages follow a multinomial distribution $\Theta_v = (\theta_1, ..., \theta_N)$. We may treat all the pages that v's out-links point to, L_v, as a sample of this hidden distribution. Similar to previous work, we collapse the same pages in L_v together, and thus the count of each page belonging to L_v is 1. A maximum likelihood estimator can then be used to estimate the Θ_v, giving us precisely the matrix M discussed in the previous section. However, the maximum likelihood estimator generates many undesirable zero probabilities due to the small size of the sample as we discussed earlier. In statistical techniques, regularization is essential when learning from small samples [Steck and Jaakkola 2002]. Here we take a Bayesian approach where regularization is achieved by specifying a *prior distribution* over the parameters and subsequently averaging over the posterior distribution [Steck and Jaakkola 2002; Heckerman et al. 1995]. That is, we put a prior distribution on the parameters Θ_v. Compared with maximum likelihood, the Bayesian approach can provide smoother estimates of the parameters.

[2] Note that T in Figure 3 (b) might be a good page. In this case, T's score should depend on *leakage* but not be dramatically boosted by its local structure.

Conjugate priors are often used in Bayesian approaches since they permit analytical calculations [Steck and Jaakkola 2002]. Specifically, we define a Dirichlet prior distribution, which is conjugate to multinomial distribution, on Θ_v with hyperparameter $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$, given by

$$Dir(\Theta_v|\alpha) = \frac{\Gamma(\sum_{i=1}^N \alpha_i)}{\Pi_{i=1}^N \Gamma(\alpha_i)} \prod_{i=1}^N \theta_i^{[\alpha_i - 1]}$$

where $\Gamma(\cdot)$ is the Gamma function. The parameters α_i can be specified by a number of approaches [Cooper and Herskovits 1992; Steck and Jaakkola 2002; Heckerman et al. 1995] and we adopt the common choice,

$$\alpha_i = \mu \cdot p(i)$$

where μ is a parameter and p is a prior distribution over all pages. Without any prior knowledge, $p(i) = P_{uniform}(i) = \frac{1}{N}$ is the uniform jumping probability[3]. Since Dirichlet is a conjugate prior for multinomial distribution, the posterior distribution

$$P(\Theta_v|L_v) \propto \prod_{i \in [1,\ldots,N]} \theta_i^{[c(i,L_v)+\mu P_{uniform}-1]}$$

is also Dirichlet, with parameters $\tilde{\alpha_i} = c(i, L_v) + \mu P_{uniform}$ where $c(i, L_v) = 1$ if $i \in L_v$ and $c(i, L_v) = 0$ otherwise. Using the fact that the Dirichlet mean is $\frac{\tilde{\alpha_i}}{\sum_k \tilde{\alpha_k}}$, we have:

$$\begin{aligned} P(i|L_v) &= \int P(i|\Theta_v) \cdot P(\Theta_v|L_v) d\Theta_v \\ &= \int \theta_i \cdot P(\Theta_v|L_v) d\Theta_v \\ &= \frac{c(i, L_v) + \mu P_{uniform}}{|L_v| + \mu} \\ &= (1 - \frac{\mu}{|L_v| + \mu}) \frac{c(i, L_v)}{|L_v|} + \frac{\mu}{|L_v| + \mu} P_{uniform} \\ &= (1 - \omega_v) P_{ml} + \omega_v P_{uniform} \end{aligned}$$

where P_{ml} is the maximum likelihood estimator and $\omega_v = \frac{\mu}{|L_v|+\mu}$. $P(i|L_v)$ is the estimated transition probability from page v to i. Note that $|L_v|$ is equal to the sum of the elements of the row corresponding to page v in A. In a Markov transition matrix form, we have:

$$\tilde{M} = diag\{1 - \omega_1, ..., 1 - \omega_N\} \cdot M + diag\{\omega_1, ..., \omega_N\} \cdot U$$

where M and U are the same as in Equation (1). The ranking scores can be calculated by solving the eigenvector equation:

$$\begin{aligned} \mathbf{r} &= \tilde{M}^T \mathbf{r} \\ &= diag\{1 - \omega_1, ..., 1 - \omega_N\} \cdot M^T \mathbf{r} + \frac{\tau}{N} \mathbf{e}_N \end{aligned}$$

[3] In general, p can reflect any prior preferences in choosing the next page. For example, p can be topic sensitive or query specific.

where τ is a weighted sum of the jumping probabilities to a random page:

$$\tau = \sum_{i=1}^{N} r(i) \times \omega_i.$$

The i-th value in vector $\mathbf{r}$ is the ranking score of the i-th web page. Since we use Dirichlet prior to calculate the ranking values, we call our algorithm *DirichletRank*. From the definition of ω_v, we can see that the larger $|L_v|$ is, the larger $1 - \omega_v$ will be. Thus in DirichletRank, a surfer would more likely follow the out-links of the current page if the page has many out-links. Intuitively, this is also reasonable. A page with more out-links is presumably a good hub page for directing surfers to good authority pages [Kleinberg 1999], and thus it is natural to believe that the surfer will follow the out-links of such pages with higher probabilities.

The derivation above is analogous to a similar derivation of the Dirichlet prior smoothing method in information retrieval [Zhai and Lafferty 2001; 2002], which has also been shown to be quite effective for retrieval.

3.2 Comparison with PageRank

Bayesian estimation provides a principled way for setting the transition probabilities. We now show that it not only solves the zero out-link problem in an elegant way, but also solves the problem of "zero-one gap" naturally.

The random jumping probability of DirichletRank is

$$\omega(n) = \frac{\mu}{n + \mu}, 0 \leq n \leq \infty \tag{6}$$

where n is number of out-links and μ is the Dirichlet prior parameter. We set $\mu = 20$ and plot $\omega(n)$ in Figure 2. The figure shows that the jumping probability in DirichletRank is smoothed and there is no big gap between 0 and 1 out-link.

To compare with the PageRank algorithm, we define $d_o(v)$ and $d_s(v)$ as the DirichletRank scores of any page v in Figure 3 (a) and (b) respectively and also assume τ of DirichletRank is the same for both Figure 3 (a) and (b) since $k \ll N$ in general. We then have the following theorem.

THEOREM 2. *With k ($k \ll N$) bogus pages, σ leakage, and τ as the weighted sum of the random jumping probabilities,*

$$d_o(T) = \sigma + \frac{\tau}{N} \tag{7}$$

$$d_s(T) = \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right]\left[\sigma + \frac{k + \mu + 1}{\mu + 1}\frac{\tau}{N}\right] \tag{8}$$

and $d_s(T) \geq d_o(T)$ for any positive integer k.

On the surface, we obtain a similar conclusion as in the PageRank scores: $d_s(T)$ is still larger than or equal to $d_o(T)$. However, $d_s(T)$ is in fact very close to $d_o(T)$. For example, when we set $\mu = 20$ and $k = 1$, $d_s(T) \approx \left[1 + \frac{1}{20^2 + 2 \times 20}\right] d_o(T) \approx d_o(T)$. This indicates that there is no significant change in T's DirichletRank scores before and after spamming. Thus DirichletRank is indeed more stable and less sensitive to the change of local structure.

We further analyze the influence of k on both PageRank and DirichletRank. In PageRank,

$$\begin{aligned} r_s(T) &= \frac{1}{2\lambda - \lambda^2}\left[\sigma + \frac{\tau(k(1-\lambda)+1)}{N}\right] \\ &= \left[\frac{1-\lambda}{2\lambda-\lambda^2}\frac{\tau}{N}\right]k + \frac{1-\lambda}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau}{N}\right] \end{aligned}$$

In DirichletRank, $1 + \frac{k}{\mu^2+(k+1)\mu} < 1 + \frac{1}{\mu}$, thus,

$$\begin{aligned} d_s(T) &= \left[1 + \frac{k}{\mu^2+(k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right] \\ &< \left[1 + \frac{1}{\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right] \\ &= \left[\frac{1}{\mu}\frac{\tau}{N}\right]k + \frac{\mu+1}{\mu}\left[\sigma + \frac{\tau}{N}\right] \end{aligned}$$

In PageRank, the scores of target pages increase linearly with the number of bogus pages k and the coefficient is $c_r = \frac{1-\lambda}{2\lambda-\lambda^2}\frac{\tau}{N}$. In DirichletRank, the scores of target pages are upper-bounded by a linear function with the coefficient $c_d = \frac{1}{\mu}\frac{\tau}{N}$. A typical setting $\lambda = 0.15$ leads to $c_r = 3.06\frac{\tau}{N}$. On the other hand, even if we set μ to the smallest value 1, $c_d \leq 1 \times \frac{\tau}{N}$ is much smaller than c_r. In fact, a typical μ, as our experiments show later, is 20, which leads to $c_d \leq 0.05\frac{\tau}{N}$. $0.05 \ll 3.06$. Thus, if a spammer creates the same number of bogus pages, the influence on DirichletRank is much less than that on PageRank.

Similar to PageRank, the major computation in DirichletRank is calculating the principal eigenvector of transition matrix $\tilde{M}$. Using the iterative Power Method as in paper [Page et al. 1999], both have similar computational cost per iteration. Note that in Equation (6), $\omega(n)$ approaches to zero as n approaches to infinity. Theoretically, when all the pages have a large number of out-links, the second eigenvalue of matrix $\tilde{M}$ in DirichletRank is close to 1, and thus the convergence rate of the Power Method is very slow [Golub and Loan 1996; Haveliwala and Kamwar 2003]. To overcome this difficulty *theoretically*, we can modify $\omega(n)$ by incorporating a small constant parameter λ:

$$\omega'(n) = \lambda + (1-\lambda)\omega(n)$$

We name the method based on $\omega'(n)$ as *TwoStageRank* ranking algorithm. However, *practically*, since the out-links of a web graph tend to follow a power law distribution, DirichletRank can converge fast. In the discussion of the experiment results, we will show that both DirichletRank and TwoStageRank can achieve comparable or even better convergence rates compared with PageRank. This means that they both are applicable to web-scale applications.

3.3 Effect on Anti-Link-Spamming

In this section, we use a typical link spam structure plotted in Figure 3 (b) to show that DirichletRank is more resistant to link spamming than PageRank.

As discussed in the previous section, a surfer in DirichletRank randomly jumps away with a higher probability if the page has fewer out-links. Therefore, a single reversed link from each B to T fails to entrap the surfer in the local spam structure. A spammer may attempt to break the DirichletRank algorithm by creating more out-links in each bogus page to point to other bogus pages. This structure would reduce the probability of jumping to a random page, and hence keep the probability mass within the local structure. However, the following theorem shows that such a spamming strategy would not work at all.

THEOREM 3. *The DirichletRank score $d(T)$ is* independent *of the internal connections between bogus pages (B's) when the following three conditions hold:*
1) Target page T has a link to each of its bogus page B;
2) Each B has a reversed link back to T;
3) T and B have no out-links to other pages except T and B.

This theorem indicates that a complex local structure may keep probability mass within a local structure, but it is not able to boost the score of target page T. In the next theorem, we show that the above situation is already the best that a spammer can do. We make the assumption that no leakage goes to bogus pages B's. This assumption is reasonable since bogus pages are created internally for boosting a target page.

THEOREM 4. *Given a fixed number of bogus pages k, when no leakage is added into any bogus page, the optimal DirichletRank score for any spamming structure is*

$$d(T) = \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right]. \qquad (9)$$

According to Theorem 3, adding any link between B's is an equivalently optimal spamming structure. Theorem 4 further claims that any additional out-links from bogus pages to the outside global web can only make a structure sub-optimal. Thus, the best that a spammer can do is to set up a spamming structure such as Figure 3 (b). However, in the previous subsection, we have demonstrated that the DirichletRank score of a target page with such an optimal spamming structure is close to the score without any spamming.

Increasing the number of bogus pages is one way to increase a target page's DirichletRank score. But as we analyzed before, the coefficient c_d of the number of bogus pages in DirichletRank is much lower, indicating a spammer needs to set up many more bogus pages. More bogus pages cost the spammer more effort, and also make spam structures much easier to detect. In the experiment section, we will empirically study the impact of number of bogus pages.

3.4 Discussions

We have thus theoretically demonstrated that DirichletRank is more resistant to several typical spam structures. In general, it is impossible to enumerate all such structures. But since most link spam structures and anti-link-spamming techniques studied in previous works are on the basis of PageRank, they are all affected by the "zero-one gap" problem. DirichletRank solves the "zero-one gap" problem, and thus can replace PageRank and provide a more sound basis. For example, the "collusion" structure studied in [Zhang et al. 2004] cannot entrap DirichletRank because the number of out-links in each target page is only 1 so that a surfer will jump away to a random page with a probability approaching to 1. We will demonstrate this in the experiment section.

A major feature of DirichletRank is that the surfer will follow the out-links of a page with higher probability, if it has more out-links. On the surface, this looks more dangerous since spam pages may have more out-links and DirichletRank can be trapped into a link farm if it is huge. However, in real web data, there are even *more* good pages which also have a large number of out-links. Our results on the real web data show that empirically DirichletRank is overall beneficial. Moreover, compared with PageRank, spammers need to pay much more effort so as to build highly-connected large link farms to manipulate the DirichletRank algorithm, while they can manipulate the PageRank algorithm with little effort by exploiting the "zero-one gap" problem.

4. EXPERIMENTS

We use two data sets in our experiments: the .GOV data set of TREC conference[4] crawled in 2002 and the .UK data set of WebGraph project[5] crawled in 2006. The .GOV data set has about fifty queries and the relevant web pages manually judged for each query. The .UK data set has a sample of web hosts which are judged by humans to be spams or not. For DirichletRank and PageRank, the .GOV data set is used to compare their search effectiveness. We use both data sets to compare their resistance against link spams (real spams on the .UK data set and simulated spams on the .GOV data set).

The .GOV data set is about 18 Gigabytes in size and contains $1,247,753$ web pages crawled from the ".gov" domain in 2002. $1,053,372$ documents are in html format; all the others are of non-html format (*e.g.*, pure text, pdf, postscript, Microsoft Word) and therefore have no out-links at all. Each page has 8.94 out-links on average. Both content and link information of pages in .GOV data set are used to study the search effectiveness.

The .UK data set is much larger. It contains 77,741,046 web pages crawled from the ".uk" domain in May, 2006. There are 2,965,197,340 hyperlinks in total in this data set and each web page has 38.14 out-links on average. We only use the link information in this data set to study the link spams; as this data set does not have queries with relevance judgments, we cannot use it to evaluate the ranking accuracy.

In the following sections, we will compare DirichletRank with PageRank in terms of ranking accuracy, stability under perturbation, and resistance against link spamming. We will show that DirichletRank achieves better retrieval accuracy than PageRank. More importantly, DirichletRank is shown to be much more stable under link perturbation and more robust against link spamming than PageRank.

4.1 Effectiveness for Web Search

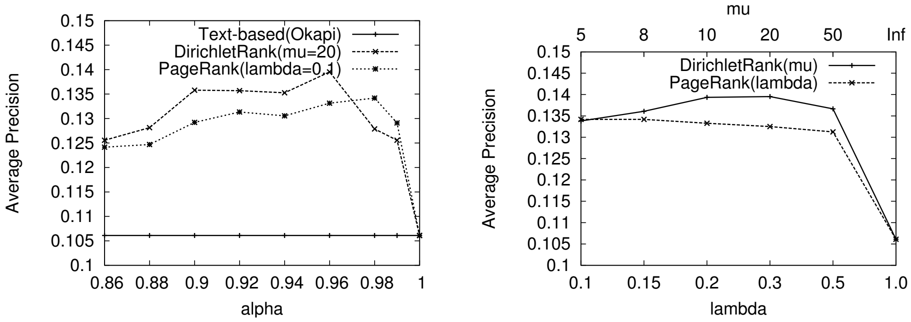
To evaluate their ranking accuracy, we use the fifty "topic distillation" topics created by NIST for TREC-2003 task. (Note that the results in this subsection are adapted from our previous work [Wang et al. 2005].) On average, there are 10.32

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

[5] http://law.dsi.unimi.it/

Table I. Results of different ranking algorithms

Methods	MAP	P@5	P@10
Okapi	0.106	0.112	0.088
PageRank (impr.)	0.134(26%)	0.148(32%)	0.11(25%)
DirichletRank (impr.)	0.140(32%)	0.156(39%)	0.116(32%)



(a) Parameter α

(b) Parameters μ and λ

Fig. 4. Performance comparison along with different parameters

relevant documents per topic. We use mean average precision (MAP), precision at 5 documents (P@5), and precision at 10 documents (P@10) as our evaluation metrics. We use the Okapi retrieval method and the BM25 weighting function for initial retrieval; the parameters are set to the same values as in [Cai et al. 2004] (*i.e.* $k1 = 4.2, k3 = 1000, b = 0.8$). We choose the top 2000 documents according to Okapi scores and construct two rankings of the documents, based on the Okapi scores and link information, respectively. We use $rank_{text}$ and $rank_{link}$ to represent the *positions* of a document in the two ranking lists. The final ranking is obtained by ordering documents increasingly according to a combination of these two ranking positions [Cai et al. 2004]:

$$\alpha \cdot rank_{text} + (1 - \alpha) \cdot rank_{link}$$

The 2000 documents are re-ranked according to the formula above and we select the top 1000 documents for the final evaluation. We vary α to select the best results for both DirichletRank and PageRank.

Table I lists the best results of PageRank and DirichletRank. Compared with the best result of TREC2003 participants (with MAP of 0.1543 and P@10 of 0.1280) [Craswell and Hawking 2003], which only used the text/html format web pages, the results of PageRank in our experiment are reasonable (with MAP of 0.134 and P@10 of 0.11). From Table I, we see that both link-based ranking algorithms improve the performance significantly over the content-based Okapi method. Furthermore, DirichletRank achieves better performance than PageRank on all the three metrics: 4.03% on MAP, 5.41% on P@5, and 5.55% on P@10. A Wilcoxon signed rank test indicates the improvement on MAP is statistically significant (p-value=0.034).

We also study the performance under different parameter settings. The results are in Figure 4. In Figure 4(a), the average precision is plotted along with different α values. DirichletRank achieves best results when $\alpha = 0.96$ and $\mu = 20$. The PageRank achieves the best result when $\alpha = 0.98$ and $\lambda = 0.1$. Figure 4(b) shows the performance under different λ and μ values where α's are set respectively to the optimal values as above. When μ goes to infinity or λ goes to 1, the link-based ranking will be uniform thus the performance is equal to the content-based method. Note that these empirical results show that a small λ value is preferred to ensure the effectiveness of PageRank; unfortunately, it makes the zero-one gap larger, causing PageRank more sensitive to link farm spams. DirichletRank achieves the best result when we set $\mu = 20$.

These results show that our solution to zero-one gap problem improves ranking accuracy over the original PageRank due to more reasonable allocation of transition probabilities. In the following experiments, we compare DirichletRank with PageRank in terms of stability under perturbation and robustness against link spamming. Unless otherwise stated, we set $\lambda = 0.15$ for PageRank as suggested in [Brin and Page 1998] and set $\mu = 20$ in the following experiments.

4.2 Stability under Perturbation

Stability is an important property for a reliable ranking algorithm. In general, a stable ranking algorithm does not change its ranking dramatically when a small perturbation (*e.g.*, removing or adding a small number of links or pages) is imposed [Ng et al. 2001]. In this section, we compare DirichletRank with PageRank in terms of stability. We simulate perturbation by varying the density of the links in the web graph in a way similar to how it is done in [Xue et al. 2005]. Specifically, we first perturb the web graph by randomly deleting $f\%$ links with f varied from 10 to 70, and then compute PageRank and DirichletRank scores in these perturbed web graphs and compare both the new ranking scores and ranking positions with the original ones.

Since PageRank/DirichletRank score vectors are the stationary probability distribution of a Markov matrix, we can measure the difference between the two score distributions by 1-norm error. Given two probability distributions $p(x)$ and $q(x)$, the 1-norm error is defined as

$$\ell_1(p,q) = \sum_x |p(x) - q(x)|$$

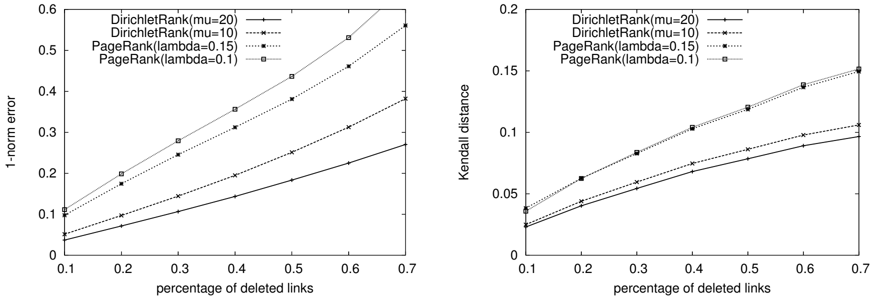
In our experiment, $p(x)$ is the scores from the original graph and $q(x)$ is the scores from the perturbed graph.

We also measure the difference between two ranking lists by normalized Kendall-τ distance. Given two ranking lists τ_1 and τ_2 of N web pages, the distance is defined as

$$Kendall(\tau_1, \tau_2) = \frac{|\{(u,v) : \tau_1, \tau_2 \text{ disagree on order of } (u,v), u \neq v\}|}{N \times (N-1)}.$$

Note that Kendall distance measures the difference based on ranking *positions*, instead of ranking *scores*.

We compare the stability of DirichletRank and PageRank in Figure 5, where the x-axis denotes the percentage of deleted links $f\%$ and y-axis denotes the 1-



(a) Measured by 1-norm error

(b) Measured by Kendall distance

Fig. 5. The stability comparison under perturbation. Small values indicate small difference.

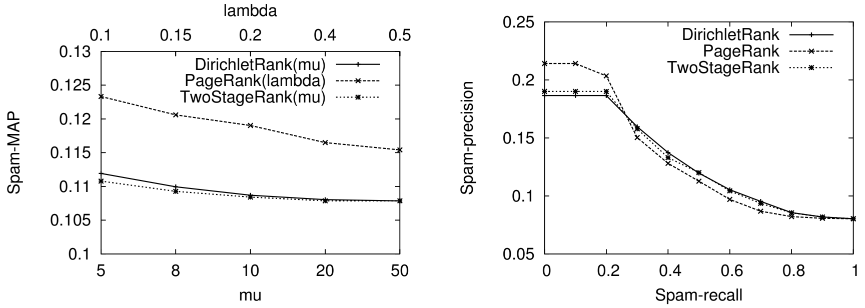
norm error in (a) and Kendall distance in (b); the smaller the value is, the more stable the algorithm is. In both figures, we can observe that the difference values of the two PageRank curves are much higher than the DirichletRank ones. This means our solution to zero-one gap makes the ranking algorithm more stable under perturbation.

4.3 Resistance against Link Spamming

In this section, we study the impact of link spamming on both DirichletRank and PageRank based on the real spams in the .UK data set and simulated spams on the .GOV data set.

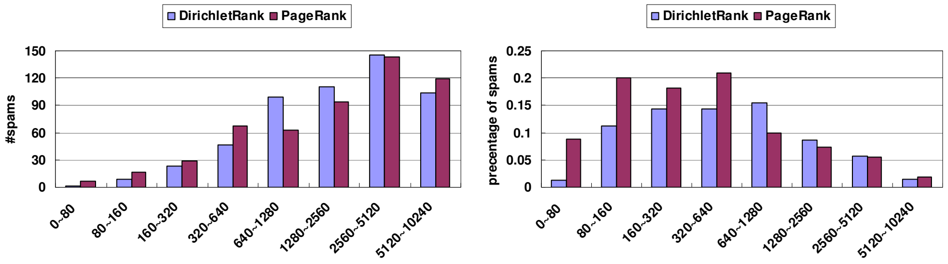
4.3.1 *Results on Real Spams.* On the .UK data set, 8415 hosts are manually labelled by humans to be spam, normal, or borderline. (More details can be found at `http://aeserver.dis.uniroma1.it/webspam`.) We only use the *spam* and *normal* hosts to construct our evaluation data set and filter out those hosts which do not have host homepages in the .UK web graph. Finally, we have 540 spam hosts and 6190 normal hosts (6730 in total) in our evaluation set. For DirichletRank or PageRank, we first compute a ranking list for the whole .UK graph. We then obtain a small ranked list of the 6730 host homepages by ignoring other pages. Taking these 6730 hosts as a sample of the whole web graph, we compare DirichletRank and PageRank using spam-precision, spam-recall, and spam-MAP defined as follows. Given a ranked list of N pages, a better or spam-resistant ranking algorithm would rank the spam pages lower in the list. To compare different ranked lists, we treat spam pages as "relevant" pages and normal pages as "irrelevant" pages. In this way, traditional precision, recall, and MAP can be defined accordingly. We name them as spam-precision, spam-recall, and spam-MAP. All these measures capture the ranking positions of the spam pages in a rank list. Different from traditional measures, lower values here indicate better performance as they mean that spam pages are ranked lower.

As we showed in Section 4.1, link information can be very useful for improving search accuracy, but its utility can be impaired by link farms. Totally ignoring link information can make search ranking algorithms immune to link farms, but at the same time, search utility would also be decreased since we do not exploit



(a) Comparison based on Spam-MAP (b) Comparison based on PR-curve

Fig. 6. Comparison of the ranking positions of spam hosts on .UK data set.



(a) Absolute number of spams (b) Relative percentage of spams

Fig. 7. Distribution of spams in different buckets.

link information. A good link analysis algorithm should be more robust against link spams while still improving search accuracy as much as possible. To compare the resistance against link spams of DirichletRank and PageRank, we vary the parameters μ of DirichletRank and λ of PageRank in their ranges where search accuracy is improved according to Section 4.1 (*i.e.*, μ is varied from 5 to 50 and λ is varied from 0.1 to 0.5). In Figure 6, we show the results evaluated by spam-MAP and spam-PR curves. In Figure 6(a) we compare them using spam-MAP. We can see that when $\mu \geq 5$, DirichletRank achieves lower spam-MAP values than all the spam-MAP values of PageRank. In Figure 6 (b), we use the *interpolated* spam-PR curves of $\mu = 20$ and $\lambda = 0.15$ to further compare DirichletRank and PageRank. Clearly, at lower spam-recall levels, the spam-precisions of DirichletRank are much lower than those of PageRank, while at other spam-recall levels, DirichletRank achieves similar spam-precisions compared with PageRank. This shows that DirichletRank can push the spam pages to the lower positions, especially from the top-ranked positions. In the Web domain, top-ranked spams are more harmful since users tend to only view the top 10 or 20 results. In this figure, we also plot the results of TwoStageRank algorithm (its λ is set to 0.05). We see that it achieves similar results to DirichletRank.

To see the distribution of the spam pages in term of ranking positions, we use a similar method as in paper [Gyongyi et al. 2004] by splitting a ranking list into

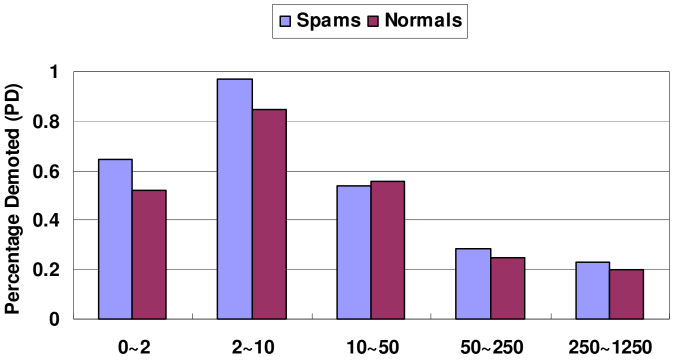


Fig. 8. Percentage of spam and normal pages demoted in different buckets.

different buckets. Each bucket contains a different number of hosts and we let the boundaries of buckets grow exponentially as we go down the rank list. Finally, we split the 6730 hosts into 8 buckets. For each bucket, we calculate both the absolute number and relative percentage of spams in it. The results are shown in Figure 7. We can see that in the first several buckets, DirichletRank has fewer spams than PageRank. For example, in the first bucket (0∼80), DirichletRank has only 1 spam page while PageRank has 7 out of 80. All these results show the effectiveness of DirichletRank on demoting spams, especially from the top-ranked positions, which is more crucial in the web domain.

Since DirichletRank emphasizes more on pages with more out-links, we empirically analyze its impact on pages with different out-degrees. We first define a measure called *relative ranking change* (RRC) for each of the 6730 pages as follows. Suppose a page i is ranked at the sth position in DirichletRank and the tth position in PageRank, its RRC is defined as

$$RRC(i) = \frac{s-t}{s+t}.$$

It is easy to verify that $-1 \leq RRC(i) \leq 1$. Inequalities $RRC(i) < 0$ and $RRC(i) > 0$ mean that page i is promoted and demoted, respectively, by DirichletRank compared with PageRank. RRC is a relative measure which emphasizes on the top-ranked positions. To show the impact of DirichletRank on pages with different out-degrees, we split all the 6730 pages into 5 buckets according to their out-degrees (*i.e.*, 0∼2, 2∼10, 10∼50, 50∼250, and 250∼1250). For each bucket, we calculate a value *percentage demoted* (PD) based on RRC for the spam and normal pages respectively as follows. Given a set of pages S, its PD is calculated as

$$PD(S) = \frac{|\sum_{i \in S, RRC(i)>0} RRC(i)|}{|\sum_{i \in S, RRC(i)>0} RRC(i)| + |\sum_{i \in S, RRC(i)<0} RRC(i)|}.$$

PD values reflect the percentage of pages demoted in terms of RRC and we use PD to show the impact of DirichletRank in Figure 8. It is not surprising to see that DirichletRank demotes more pages with lower out-degrees. For example, in bucket 2∼10, more than 80% of pages are demoted while in bucket 50∼250, only around

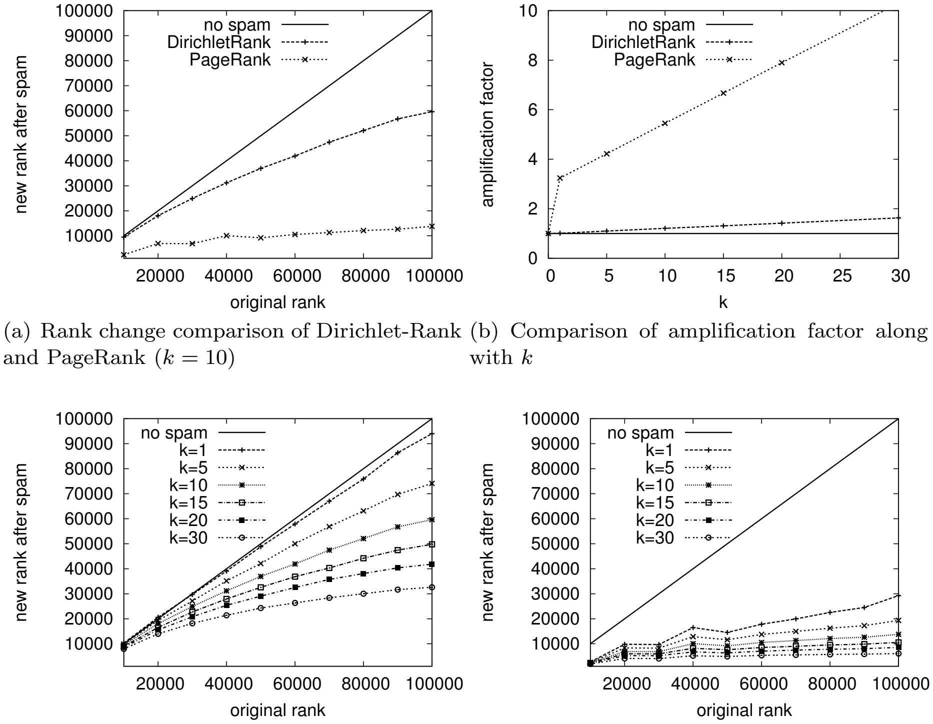
20% of pages are demoted. An interesting observation is that: in most buckets, by comparing the PD's of spam and normal pages, we can see that DirichletRank demotes more spam pages than normal ones. This shows that the emphasis on pages with many out-links by DirichletRank is overall beneficial.

4.3.2 *Results on Simulated Bogus-Page-Based Spams.* We further study the impact of bogus pages on DirichletRank and PageRank by simulating bogus-page-based spams on the .GOV data set. For both DirichletRank and PageRank, we use the same way to simulate the spams as follows: Given a link-based ranking algorithm, we first calculate the baseline spam-free ranking scores using the original clean .GOV data. We then select ten pages (*i.e.*, the $10,000$th, $20,000$th, ..., and $100,000$th) from the spam-free ranking list as our target pages. We remove the out-links of all these target pages, and for each target page, create k bogus pages, each with an in-link from and an out-link to the corresponding target page. After spamming these target pages, we re-calculate the ranking scores for all the pages on this spammed .GOV data set. We evaluate the vulnerability of the ranking algorithm by comparing the *ranking (position) change* of each target page before and after spamming and by computing the amplification factor, which is defined as the ratio of the new *ranking score* to the old *ranking score* as in paper [Zhang et al. 2004].

Figure 9 shows the changes of the 10 spammed target pages. Note that the comparisons *only* apply to *the 10 spammed pages.* We set $k = 10$ and plot two curves in Figure 9(a) to compare the rank changes of DirichletRank and PageRank. To facilitate comparison, we also plot the straight diagonal line, which represents the ideal case when no rank has changed; clearly, the closer a curve is to the diagonal line, the less sensitive the corresponding ranking algorithm is. We observe in Figure 9(a) that the DirichletRank curve is much closer to the diagonal line. For example, after spamming, the $50,000$th page is promoted to the $9,201$th by PageRank, but only to the $36,940$th by DirichletRank. This confirms that DirichletRank is much less sensitive to bogus-page-based spamming. Figure 9(b) shows the average amplification factor of the ten target pages along with k. In both algorithms, the amplification factors increase roughly linearly with k, but DirichletRank has a nearly flat slope and significantly lower amplification factor values. Note that there is a jump between $k = 0$ and $k = 1$ in the PageRank curve, precisely because of the "zero-one gap" problem that we discussed at the beginning of this paper. This supports Theorem 1 and Theorem 2 empirically.

Figure 9 (c) and (d) show the ranking position changes of the 10 spammed pages in DirichletRank and PageRank respectively. It is expected that a larger k would promote a target page more than a smaller k, but we can also clearly see that curves in Figure 9(c) are much closer to the diagonal line, indicating that DirichletRank is significantly less sensitive than PageRank for all the k values plotted. Indeed, the impact of 30 bogus pages on DirichletRank is still much less than that of a single bogus page on PageRank.

4.3.3 *Results on Simulated Collusion Spams.* We now study the impact of the link alliance spam structures, in which a group of spammers collaborate to build link farms [Gyongyi and Garcia-Molina 2005a]. In particular, we study the collu-



(a) Rank change comparison of Dirichlet-Rank and PageRank ($k = 10$)

(b) Comparison of amplification factor along with k

(c) Impact of number of bogus pages k on DirichletRank

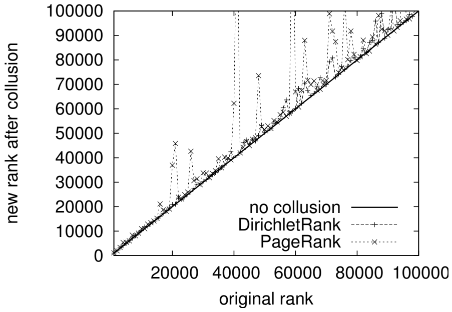
(d) Impact of number of bogus pages k on PageRank

Fig. 9. Resistance comparison on bogus-page-based spams. Note that the comparisons *only* apply to *the 10 spammed pages.*

sion structure identified in [Zhang et al. 2004], in which a set of nodes modify their out-links to improve each other's PageRank scores. In [Zhang et al. 2004], collusion structures are detected by predefined rules, which needs to calculate PageRank eigenvectors multiple times for several different λ values, and thus it is time consuming. We find that our DirichletRank algorithm can solve the collusion problem well without computing the eigenvectors multiple times.

Similar to [Zhang et al. 2004], we select 100 ranking positions, 1000th, 2000th, ..., and 100000th. At each position, we select two adjacent pages, delete all their out-links, and then add two links between them with each pointing to the other. We calculate the rankings before and after the modification.

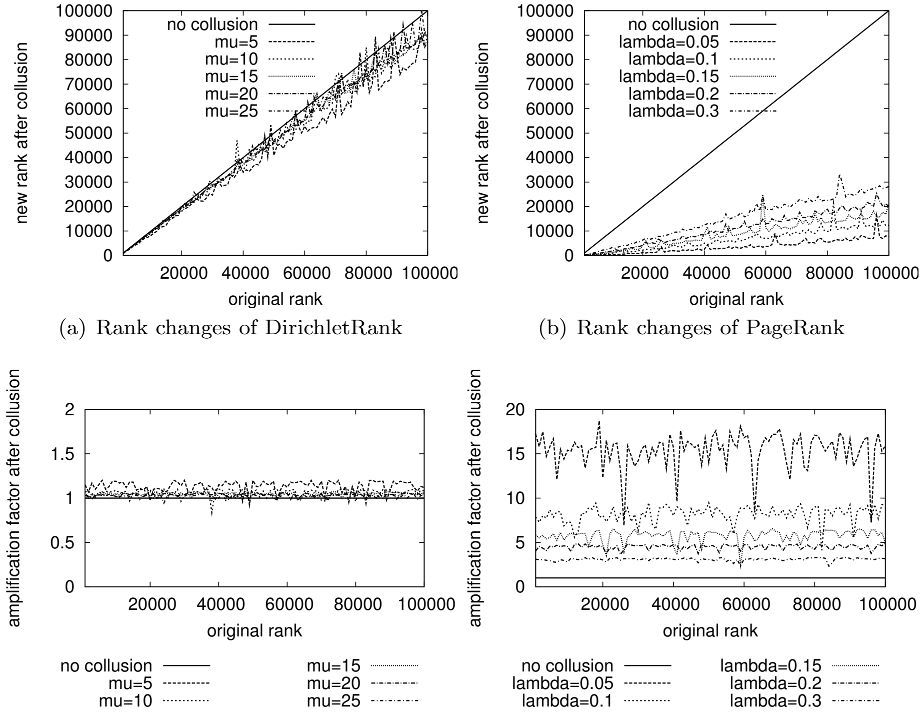
Figure 10 shows the impact of the simulated collusions. Note that the comparisons *only* apply to *the spammed pages.* We show both the results when we delete the out-links and when we create the collusions for the 100 pairs of pages. Once again, we see clearly that such collusions can not change the ranking of DirichletRank algorithm much, but can dramatically change the PageRank ranking. Figure 11 further shows the impact of collusions under different values of λ and μ. We again observe that PageRank is very sensitive to collusion structures, while DirichletRank is much more stable. For example, the amplification factor in PageRank is up to 20 when



(a) When deleting the out-links

(b) When creating the collusions

Fig. 10. Comparison of DirichletRank and PageRank on collusions. Note that the comparisons *only* apply to *the 100 spammed pairs*



(a) Rank changes of DirichletRank

(b) Rank changes of PageRank

(c) Amplification factor of DirichletRank

(d) Amplification factor of PageRank

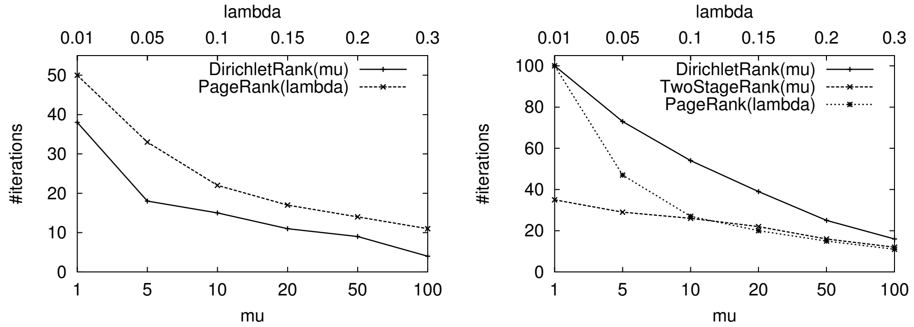
Fig. 11. Parameter influences of DirichletRank and PageRank after collusion spams. Note that the comparisons *only* apply to *the spammed pages*.

$\lambda = 0.05$, but all the amplification factor values are close to 1 in DirichletRank.

4.3.4 *Degradation of Retrieval Accuracy from Spams.* To study the impact of link spams on retrieval accuracy, we use the same queries and the same .GOV data set as Section 4.1 in this experiment. We simulate the link spams by randomly

Table II. The impact of spams on retrieval accuracy

Methods	MAP	P@5	P@10
PageRank	0.134	0.148	0.11
Spammed PageRank (diff.)	0.123(−8%)	0.132(−11%)	0.102(−7%)
DirichletRank	0.140	0.156	0.116
Spammed DirichletRank (diff.)	0.139(−1%)	0.156(−0%)	0.116(−0%)



(a) Number of iterations on .GOV data set (b) Number of iterations on .UK data set

Fig. 12. Comparison of computational complexity.

selecting 5% from the pages which contain query keywords and randomly creating the spam structures studied above (5 bogus pages or collusion) for them. The results are summarized in Table II as spammed PageRank and spammed DirichletRank. Compared with DirichletRank, PageRank is very sensitive to the spam structures and the accuracies are deteriorated much. DirichletRank is much more robust and its accuracies are almost the same as the one without spamming. All the results confirm that DirichletRank is much more resistant against link spamming by addressing the zero-one gap problem.

4.4 Time Complexity and Score Distribution

Using Power Method, DirichletRank and PageRank have similar complexity for a single iteration. Therefore, the total computation time is determined by the number of iterations needed for convergence. In Figure 12, we show the number of iterations needed for convergence for both DirichletRank and PageRank. (We stop the iterations when the 1-norm error of score vectors in the last two iterations is less than 0.001.) On both data sets, the numbers of iterations decrease as the parameter value increases. This is because the weighted sum of the random jumping probabilities τ increases as the parameter value increases. It can be seen that on .GOV data set, DirichletRank converges faster than PageRank, while on .UK data set, PageRank converges faster than DirichletRank. This is because the average number of out-links of .UK data set is larger than that of .GOV data set and the τ values are smaller on the .UK data set for DirichletRank. When $\lambda = 0.15$ and $\mu = 20$, the numbers of iterations of DirichletRank and PageRank are comparable. In Figure 12(b), the results of TwoStageRank (its $\lambda = 0.05$) are also plotted. Clearly, TwoStageRank can converge faster than DirichletRank and even faster

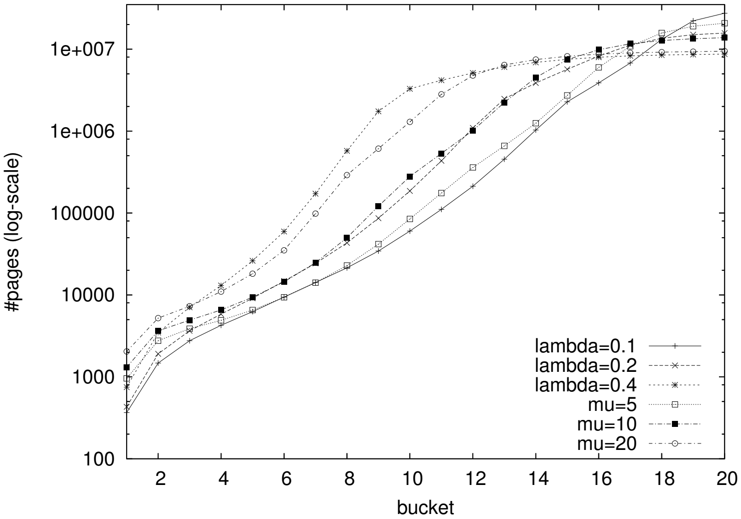


Fig. 13. The distribution of the number of pages with respect to 20 buckets.

than PageRank on the .UK data set, which confirms our discussion in Section 3.2.

To show the distribution of ranking scores, we partition the pages into 20 buckets as follows: Given a ranking list of pages in decreasing order of their ranking scores, we partition the list into buckets such that the sum of scores of each bucket is 0.05. Thus the first bucket contains those pages which have highest scores. Figure 13 shows the results for different parameters and we can see that all the lines have similar slopes before bucket 14 and the lines tend to be flat after bucket 14 when the parameters (both μ and λ) are larger. We can see that the shapes of the curves are related to the parameters μ and λ. Furthermore, with parameters such as $\mu = 10$ and $\lambda = 0.2$, Figure 13 shows that DirichletRank and PageRank can have similar skewness, and thus similar score distribution.

5. RELATED WORK

PageRank [Brin and Page 1998] and HITS [Kleinberg 1999] are the two earliest link analysis algorithms using eigenvectors to identify "authoritative" pages via hyperlink information. Since the introduction of these two algorithms, several methods have been developed to improve their accuracy. BHITS [Bharat and Henzinger 1998] tries to alleviate the dominance of certain special link structures in the HITS algorithm. ARC [Chakrabarti et al. 1998] combines the pure link-based HITS with the anchor text describing the topics. *Randomized HITS* and *Subspace HITS* [Ng et al. 2001] modify HITS to get a more stable algorithm. PHITS [Cohn and Chang 2000] proposes a statistical hubs and authorities algorithm. [Li et al. 2002] solves the "small-in-large-out" problem of HITS. [Haveliwala 2002] calculates topic-sensitive PageRanks based on the web page categories. *Focused PageRank* and *Double Focused PageRank* [Diligenti et al. 2002] and also the *Intelligent Surfer* [Richardson and Domingos 2002] try to combine the link and content information in PageRank. The visual block structures within web pages and the directory structures of web sites are used to improve PageRank accuracy by [Cai et al. 2004] and [Xue et al. 2005] respectively. The stability of the link analysis algorithms are studied in [Ng et al. 2001; Borodin et al. 2001; 2005; Lempel and Moran 2005] and PageRank is shown to be relatively more stable than HITS in general [Ng et al. 2001]. A flurry of works are studied to conduct theoretical analysis on PageRank and accelerate its computation. For example, [Baeza-Yates et al. 2006] studies PageRank from the perspective of damping functions and it generalizes the original PageRank algorithm by proposing several different types of damping functions. Other examples include [Kamvar et al. 2003; Boldi et al. 2005; Bianchini et al. 2005; Jeh and Widom 2003; McSherry 2005]. A good survey of PageRank can be found in [Berkhin 2005]. However, to the best of our knowledge, none of the work addressed the "zero-one gap" problem of PageRank.

Web spamming is studied recently. The taxonomy of web spamming is defined in [Gyongyi and Garcia-Molina 2005b]. Link alliance structures, in which a group of spammers collaborate to build link farms, are studied in [Gyongyi and Garcia-Molina 2005a] theoretically. However, detection of web spamming, especially link spamming, is a very challenging problem. Most previous anti-spamming work addresses it heuristically. For example, [Fetterly et al. 2004] analyzes the the statistical distribution of web pages and treats the outliers as spam pages. [Zhang et al. 2004] observes that the influence of spam structures is sensitive to the random jumping probability λ and hence proposes two heuristics of personalizing λ for spam detection. [Benczur et al. 2005] assumes that only honest pages have their in-link neighbors' PageRank scores obeying a power law distribution. This method, however, would not work well when a page does not have a sufficiently large number of in-link neighbors. Moreover, it is computationally expensive, and thus cannot handle large data sets efficiently. [Wu and Davison 2005] identifies a link farm seed set through a page's out-link and in-link domains, and then makes a second expansion to identify more spams. It is very sensitive to the parameter settings. [Ntoulas et al. 2006] and [Becchetti et al. 2006] use classification techniques to detect spams relying on content analysis and link analysis respectively.

TrustRank [Gyongyi et al. 2004] and Topical TrustRank [Wu et al. 2006] are two PageRank-like methods proposed recently to quantify pages' honesties. Both rely on the fact that good pages seldom point to bad ones and propagates trust scores through hyperlinks from good seeds, which are identified by human experts manually. These methods incorporate human labeling and thus could be more reliable than the above heuristics. However, since they are biased PageRanks, both suffer from the "zero-one gap" problem and thus could be ineffective to anti-spamming.

The current work is an extension of our previous work [Wang et al. 2005]. In this work, we show that our method addresses a fundamental problem in PageRank that makes PageRank prone to be spammed. Since most existing methods are based on PageRank, our method can potentially be combined with them to improve their effectiveness in anti-spamming.

6. CONCLUSIONS

In this paper, we showed that the popular link-based ranking algorithm PageRank has a "zero-one gap" problem, which can be potentially exploited to spam its ranking results. We propose a novel DirichletRank algorithm, which computes these probabilities in a more principled way using Bayesian estimation with a Dirichlet prior. Experiment results show that DirichletRank improves ranking accuracy over the original PageRank due to its more reasonable allocation of transition probabilities. More importantly, we show both analytically and empirically that DirichletRank is much more robust against link spamming than PageRank. We compared the stability of DirichletRank and PageRank under perturbation and evaluated the influence of the real spams, simulated bogus-page-based spams, and simulated collusion spams. In all the experiments, DirichletRank is shown to be more stable under perturbation and be substantially more resistant against link spamming. Since most existing anti-spamming methods are based on PageRank, our method can potentially be combined with them to improve their effectiveness. DirichletRank can be computed as efficiently as PageRank by Power Method, and thus it is scalable to large-scale web applications.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. We also thank Xin He, Jing Jiang, and Hui Fang for their useful discussions. This work is in part supported by the National Science Foundation under award numbers IIS-0347933 and IIS-0713581.

REFERENCES

Baeza-Yates, R. A., Boldi, P., and Castillo, C. 2006. Generalizing PageRank: Damping Functions for Link-based Ranking Algorithms. In *SIGIR*. 308–315.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., and Baeza-Yates, R. 2006. Link-Based Characterization and Detection of Web Spam. In *Second International Workshop on Adversarial Information Retrieval on the Web*.

Benczur, A., Csalogany, K., Sarlos, T., and Uher, M. 2005. SpamRank-Fully Automatic Link Spam Detection. In *First International Workshop on Adversarial Information Retrieval on the Web*.

Berkhin, P. 2005. A Survey on PageRank Computing. *Internet Mathematics 1,* 3, 335–380.

Bharat, K. and Henzinger, M. R. 1998. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *SIGIR*. 104–111.

Bianchini, M., Gori, M., and Scarselli, F. 2005. Inside PageRank. *ACM Transaction of Internet Technology 5,* 1, 92–128.

Boldi, P., Santini, M., and Vigna, S. 2005. PageRank as a Function of the Damping Factor. In *WWW*. 557–566.

Borodin, A., Roberts, G. O., Rosenthal, J. S., and Tsaparas, P. 2001. Finding Authorities and Hubs from Link Structures on the World Wide Web. In *WWW*. 415–429.

Borodin, A., Roberts, G. O., Rosenthal, J. S., and Tsaparas, P. 2005. Link Analysis Ranking: Algorithms, Theory, and Experiments. *ACM Transaction of Internet Technology 5,* 1, 231–297.

Brin, S. and Page, L. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks 30,* 1-7, 107–117.

Cai, D., He, X., Wen, J.-R., and Ma, W.-Y. 2004. Block-Level Link Analysis. In *SIGIR*. 440–447.

Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., and Rajagopalan, S. 1998. Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text. In *WWW*. 65–74.

Cohn, D. and Chang, H. 2000. Learning to Probabilistically Identify Authoritative Documents. In *ICML*. 167–174.

Cooper, G. F. and Herskovits, E. 1992. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning 9*, 309–347.

Craswell, N. and Hawking, D. 2003. Overview of the TREC 2003 Web Track. In *The Twelfth Text Retrieval Conference (TREC 2003)*. 78–92.

Diligenti, M., Gori, M., and Maggini, M. 2002. Web Page Scoring Systems for Horizontal and Vertical Search. In *WWW*. 508–516.

Ding, C. H. Q., He, X., Husbands, P., Zha, H., and Simon, H. D. 2002. PageRank: HITS and a Unified Framework for Link Analysis. Technical report, LBNL.

Eiron, N., McCurley, K. S., and Tomlin, J. A. 2004. Ranking the Web Frontier. In *WWW*. 309–318.

Fetterly, D., Manasse, M., and Najork, M. 2004. Spam, Damn Spam, and Statistics: Using Statistical Analysis to Locate Spam Web Pages. In *WebDB*. 1–6.

Golub, G. H. and Loan, C. F. V. 1996. *Matrix Computations*. The Johns Hopkins University Press, Baltimore.

Gyongyi, Z. and Garcia-Molina, H. 2005a. Link Spam Alliances. In *VLDB*. 517–528.

Gyongyi, Z. and Garcia-Molina, H. 2005b. Web Spam Taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*.

Gyongyi, Z., Garcia-Molina, H., and Pedersen, J. 2004. Combating Web Spam with TrustRank. In *VLDB*. 576–587.

Haveliwala, T. H. 2002. Topic-Sensitive PageRank. In *WWW*. 517–526.

Haveliwala, T. H. and Kamwar, S. D. 2003. The Second Eigenvalue of the Google Matrix. Technical report, Stanford University. March.

Heckerman, D., Geiger, D., and Chickering, D. M. 1995. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning 20,* 3, 197–243.

Jeh, G. and Widom, J. 2003. Scaling Personalized Web Search. In *WWW*. 271–279.

Kamvar, S. D., Haveliwala, T. H., Manning, C. D., and Golub, G. H. 2003. Extrapolation Methods for Accelerating PageRank Computations. In *WWW*. 261–270.

Kleinberg, J. M. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM 46,* 5, 604–632.

Lempel, R. and Moran, S. 2005. Rank-Stability and Rank-Similarity of Link-Based Web Ranking Algorithms in Authority-Connected Graphs. *Information Retrieval 8,* 2, 245–264.

Li, L., Shang, Y., and Zhang, W. 2002. Improvement of HITS-based Algorithms on Web Documents. In *WWW*. 527–535.

McSherry, F. 2005. A Uniform Approach to Accelerated PageRank Computation. In *WWW*. 575–582.

Motwani, R. and Raghavan, P. 1995. *Randomized Algorithms*. Cambridge University Press, United Kingdom.

Ng, A. Y., Zheng, A. X., and Jordan, M. I. 2001. Stable Algorithms for Link Analysis. In *SIGIR*. 258–266.

Ntoulas, A., Najork, M., Manasse, M., and Fetterly, D. 2006. Detecting Spam Web Pages through Content Analysis. In *WWW*. 83–92.

Page, L., Brin, S., Motwani, R., and Winograd, T. 1999. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford University. November.

Richardson, M. and Domingos, P. 2002. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *NIPS*. 1441–1448.

Steck, H. and Jaakkola, T. 2002. On the Dirichlet Prior and Bayesian Regularization. In *NIPS*. 697–704.

Wang, X., Shakery, A., and Tao, T. 2005. Dirichlet PageRank. In *SIGIR*. 661–662.

Wu, B. and Davison, B. D. 2005. Identifying Link Farm Spam Pages. In *WWW (Special interest tracks and posters)*. 820–829.

Wu, B., Goel, V., and Davison, B. D. 2006. Topical TrustRank: Using Topicality to Combat Web Spam. In *WWW*. 63–72.

Xue, G.-R., Yang, Q., Zeng, H.-J., Yu, Y., and Chen, Z. 2005. Exploiting the Hierarchical Structure for Link Analysis. In *SIGIR*. 186–193.

Zhai, C. and Lafferty, J. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR*. 334–342.

Zhai, C. and Lafferty, J. 2002. Two-Stage Language Models for Information Retrieval. In *SIGIR*. 49–56.

Zhang, H., Goel, A., Govindan, R., Mason, K., and Roy, B. V. 2004. Improving Eigenvector-Based Reputation Systems Against Collusions. In *Workshop on Algorithms and Models for the Web Graph*.

A. PROOFS

A.1 Proof of Theorem 1

Proof. Equation (4) follows directly from the definition of PageRank, and Equation (5) is proven in [Gyongyi and Garcia-Molina 2005a]. Here we prove $r_s(T) \geq \frac{1}{2\lambda-\lambda^2} r_o(T)$ as follows:

$$\begin{aligned} r_s(T) &= \frac{1}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau(k(1-\lambda)+1)}{N}\right] \\ &\geq \frac{1}{2\lambda-\lambda^2}\left[\sigma + \frac{\tau(2-\lambda)}{N}\right] \\ &= \frac{1}{2\lambda-\lambda^2} r_o(T). \end{aligned}$$

□

A.2 Proof of Theorem 2

Proof. Equation (7) follows directly from the definition of DirichletRank. Here we only show the proof of Equation (8).
According to the definition of DirichletRank, we have

$$d_s(T) = \sigma + \frac{1}{1+\mu}\sum_{\text{all B's}} d_s(B) + \frac{\tau}{N} \quad (10)$$

$$d_s(B) = \frac{1}{k+\mu} d_s(T) + \frac{\tau}{N}, \text{for all bogus pages.} \quad (11)$$

Replacing $d_s(B)$ in (10) by (11) yields

$$d_s(T) = \sigma + \frac{k}{(1+\mu)(k+\mu)} d_s(T) + \left[1 + \frac{k}{\mu+1}\right]\frac{\tau}{N}.$$

By straightforward algebra, we have

$$d_s(T) = \left[1 + \frac{k}{\mu^2+(k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right].$$

Observing that $\frac{k}{\mu^2+(k+1)\mu} > 0$ and $\frac{k+\mu+1}{\mu+1} > 1$, we have

$$d_s(T) > \sigma + \frac{\tau}{N} = d_o(T),$$

and this proves (8). □

A.3 Proof of Theorem 3

Proof. Let us assume that the ith bogus page has $n_i - 1$ out-links to other bogus pages and a link back to the target page. According to the definition of DirichletRank, we have

$$d(T) = \sigma + \sum_{i=1}^{k} \frac{d(B_i)}{n_i + \mu} + \frac{\tau}{N} \tag{12}$$

and each bogus page B_i has the score

$$d(B_i) = \frac{d(T)}{k + \mu} + \sum_{j:j \to i} \frac{1}{n_j + \mu} d(B_j) + \frac{\tau}{N}$$

where $j \to i$ denotes that B_j has a link to B_i.
Summing all $d(B_i)$ together yields:

$$\begin{aligned} \sum_{i=1}^{k} d(B_i) &= \frac{k}{k+\mu} d(T) + \sum_{i=1}^{k} \frac{n_i - 1}{n_i + \mu} d(B_i) + k\frac{\tau}{N} \\ \Rightarrow \sum_{i=1}^{k} (1 - \frac{n_i - 1}{n_i + \mu}) d(B_i) &= \frac{k}{k+\mu} d(T) + k\frac{\tau}{N} \\ \Rightarrow \sum_{i=1}^{k} \frac{d(B_i)}{n_i + \mu} &= \frac{k}{(k+\mu)(1+\mu)} d(T) + \frac{k}{1+\mu}\frac{\tau}{N} \end{aligned} \tag{13}$$

Replacing $\sum_{i=1}^{k} \frac{d(B_i)}{n_i+\mu}$ in (12) by (13), we have

$$d(T) = \left[1 + \frac{k}{\mu^2 + (k+1)\mu}\right] \left[\sigma + \frac{k + \mu + 1}{\mu + 1} \frac{\tau}{N}\right].$$

Clearly this equation is independent of the values of n_i, which proves Theorem 3. □

A.4 Proof of Theorem 4

Proof. Let us assume T has n out-links and B_i has n_i out-links where $n, n_i \geq 0$. We also assume l out of n out-links of T and l_i out of n_i out-links of B_i point to the pages in the local structure: T or other B's ($0 \leq l \leq n$, $0 \leq l_i \leq n_i$, and $l, l_i \leq k$), and others point to outside. According to the definition of DirichletRank, we have:

$$d(T) = \sigma + \sum_{i:i \to T} \frac{d(B_i)}{n_i + \mu} + \frac{\tau}{N} \tag{14}$$

We use $\to$ ($\nrightarrow$) to denote that there is (not) a link between two pages. For each bogus page B_i, its score is

$$d(B_i) = \begin{cases} \frac{d(T)}{n+\mu} + \sum_{j:j \to i} \frac{d(B_j)}{n_j+\mu} + \frac{\tau}{N} & \text{if } T \to i \\ \sum_{j:j \to i} \frac{d(B_j)}{n_j+\mu} + \frac{\tau}{N} & \text{if } T \nrightarrow i \end{cases}$$

Summing all the $d(B_i)$ together yields:

$$\begin{aligned}\sum_{i=1}^{k} d(B_i) &= \frac{l}{n+\mu}d(T) + \sum_{i:i\to T}\frac{l_i-1}{n_i+\mu}d(B_i)\\ &+ \sum_{i:i\not\to T}\frac{l_i}{n_i+\mu}d(B_i) + k\frac{\tau}{N} \end{aligned} \tag{15}$$

By splitting $\sum_{i=1}^{k} d(B_i) = \sum_{i:i\to T} d(B_i) + \sum_{i:i\not\to T} d(B_i)$, Equation (15) leads to:

$$\begin{aligned}&\sum_{i:i\to T}\frac{(n_i-l_i)+(1+\mu)}{n_i+\mu}d(B_i)\\ &= \frac{l}{n+\mu}d(T) - \sum_{i:i\not\to T}\frac{n_i-l_i+\mu}{n_i+\mu}d(B_i) + k\frac{\tau}{N}\\ &\le \frac{l}{n+\mu}d(T) + k\frac{\tau}{N}\end{aligned} \tag{16}$$

Since $l_i \le n_i$, we have

$$\sum_{i:i\to T}\frac{(n_i-l_i)+(1+\mu)}{n_i+\mu}d(B_i) \ge \sum_{i:i\to T}\frac{1+\mu}{n_i+\mu}d(B_i) \tag{17}$$

Since $l \le n$ and $l \le k$, we have

$$\frac{l}{n+\mu}d(T) \le \frac{l}{l+\mu}d(T) \le \frac{k}{k+\mu}d(T) \tag{18}$$

By combining inequality (16), (17), and (18), we obtain

$$\begin{aligned}&\sum_{i:i\to T}\frac{1+\mu}{n_i+\mu}d(B_i) \le \frac{k}{k+\mu}d(T) + k\frac{\tau}{N}\\ \Rightarrow &\sum_{i:i\to T}\frac{d(B_i)}{n_i+\mu} \le \frac{k}{(k+\mu)(1+\mu)}d(T) + \frac{k}{1+\mu}\frac{\tau}{N}\end{aligned} \tag{19}$$

Further replacing $\sum_{i:i\to T}\frac{d(B_i)}{n_i+\mu}$ in Equation (14) by the inequity (19) yields

$$\begin{aligned}&d(T) \le \sigma + \frac{k}{(k+\mu)(1+\mu)}d(T) + \frac{k}{1+\mu}\frac{\tau}{N} + \frac{\tau}{N}\\ \Rightarrow &\left[1 - \frac{k}{(k+\mu)(1+\mu)}\right] d(T) \le \sigma + \frac{k+\mu+1}{1+\mu}\frac{\tau}{N}\\ \Rightarrow &\ d(T) \le \left[1 + \frac{k}{\mu^2+(k+1)\mu}\right]\left[\sigma + \frac{k+\mu+1}{\mu+1}\frac{\tau}{N}\right].\end{aligned}$$

□

Received Month Year; revised Month Year; accepted Month Year