

Dirichlet PageRank

Xuanhui Wang, Azadeh Shakery, Tao Tao
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801
{xwang20, shakery, taotao}@cs.uiuc.edu

ABSTRACT

PageRank has been known to be a successful algorithm in ranking web sources. In order to avoid the rank sink problem, PageRank assumes that a surfer, being in a page, jumps to a random page with a certain probability. In the standard PageRank algorithm, the jumping probabilities are assumed to be the same for all the pages, regardless of the page properties. This is not the case in the real world, since presumably a surfer would more likely follow the out-links of a high-quality hub page than follow the links of a low-quality one. In this poster, we propose a novel algorithm “Dirichlet PageRank” to address this problem by adapting flexible jumping probabilities based on the number of out-links in a page. Empirical results on TREC data show that our method outperforms the standard PageRank algorithm.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Search Process

General Terms: Algorithms

1. INTRODUCTION

The PageRank algorithm [1] computes the importance scores of web pages through a stochastic irreducible Markov transition matrix \tilde{M} which is constructed from all hyperlinks between web pages. Directly defining \tilde{M} as the normalized adjacency matrix M of the web graph always produces a sparse and reducible matrix, yielding to the “rank sink” problem [1]. To solve this problem, [1] introduces a uniform matrix U ($U_{ij} = \frac{1}{N}$ where N is the number of the web pages considered.), and linearly interpolates it with M : $\tilde{M} = (1 - \lambda) \cdot M + \lambda \cdot U$. The intuition is that a web surfer will follow the out-links of the current page with probability $1 - \lambda$ and will jump to a random page with probability λ .

The standard PageRank algorithm uses a fixed λ for every web page. This may not be the best choice. As studied in HITS algorithm [3], every web page is assigned two quality scores: *hub* and *authority*. A good hub on a topic is to direct the users to authoritative pages on that topic. Thus, presumably a surfer tends to follow the out-links of a high-quality hub page rather than a low-quality one. This motivates us to think that a dynamic λ value based on the page properties can be a better choice.

Interestingly, we observe that the format of the interpolation formula $\tilde{M} = (1 - \lambda) \cdot M + \lambda \cdot U$ is very similar to the

smoothing method in language models for information retrieval [4]. In these models, a Dirichlet smoothing is proved to be more effective than a fixed λ Jelinek-Mercer smoothing because a Dirichlet smoothing can dynamically adjust the interpolation parameter according to the document length. Driven by this similarity, we hypothesize that a Dirichlet dynamic setting of interpolation parameter λ can model a PageRank Markov matrix more accurately. Unlike the language model smoothing problem, a web page length does not carry much information in the PageRank matrix, thus we would rather consider the number of out-links of a page. A page with more out-links is more likely to be a good hub page than a page with fewer ones, thus we assign a lower λ to this page. This assignment assumes a surfer would follow the out-links of the pages which are better potential hub pages and thus could model its behaviors more accurately.

2. DIRICHLET PAGERANK

In the surfer model, we would like to estimate the probability that a surfer will visit a page in the next step, either through the out-links of the current page or randomly jumping. Given a collection of N web pages, assume that A is the adjacency matrix of the corresponding web graph and that L_p denotes all the out-links contained in the page p . Also assume that a surfer selects the next page using a multinomial distribution with parameter Θ_p and that L_p is the observed data. We can find the maximum likelihood estimator by normalizing the corresponding row in A to sum to 1 (The obtained matrix is the same as M above). However, the maximum likelihood estimator is not accurate because the unseen links will get zero probabilities. To estimate the surfer model more accurately, we use a Dirichlet prior on Θ_p with hyperparameter $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)$, given by

$$Dir(\Theta|\alpha) = C(\alpha) \prod_{i=1}^N \theta_i^{\alpha_i - 1}$$

where $C(\alpha)$ is the normalization factor. The parameters α_i are chosen to be $\alpha_i = \mu \cdot P_{rand}$ where μ is a parameter and $P_{rand} = \frac{1}{N}$ is the uniform jumping probability. The Bayesian estimate of the probability that a surfer will select link l after page p is

$$P(l|L_p) = \int P(l|\Theta_p)P(\Theta_p|L_p)d\Theta_p$$

where the posterior probability is given by

$$P(\Theta_p|L_p) \propto \prod_{\text{all links } l} P(l|\Theta_p)^{c(l,L_p) + \mu P_{rand} - 1}$$

Table 1: Results of different PageRanks

Methods	AvgPrec	P@10
Text-based	0.106	0.088
PageRank (impr.)	0.134(26.4%)	0.11(25%)
Dirichlet PR (impr.)	0.140(31.5%)	0.116(31.8%)

and so is also Dirichlet, with parameters $\alpha_i = c(l, L_p) + \mu P_{rand}$ where $c(l, L_p)$ is the number of times link l appears in L_p . Using the fact that the Dirichlet mean is $\frac{\alpha_i}{\sum_k \alpha_k}$, we will have:

$$\begin{aligned}
 P(l|L_p) &= \frac{c(l, L_p) + \mu P_{rand}}{|L_p| + \mu} \\
 &= \left(1 - \frac{\mu}{|L_p| + \mu}\right) \frac{c(l, L_p)}{|L_p|} + \frac{\mu}{|L_p| + \mu} P_{rand} \\
 &= (1 - \omega_p) P_{ml} + \omega_p P_{rand}
 \end{aligned}$$

where P_{ml} is the maximum likelihood estimator and $\omega_p = \frac{\mu}{|L_p| + \mu}$. Note that $|L_p|$ equals to the sum of the elements of the row corresponding to page p in A . In a Markov transition matrix form, we have

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

The PageRank values can be calculated by solving the eigenvector equation:

$$\mathbf{v}^T \tilde{M} = \mathbf{v}^T$$

The i -th value in the vector \mathbf{v} is the PageRank score of the i -th web page. Since we use Dirichlet prior to calculate the PageRank values, we call our algorithm ‘‘Dirichlet PageRank’’. From the definition of ω_p , we can see that the larger the $|L_p|$ is, the larger $1 - \omega_p$ will be. Thus in Dirichlet PageRank, a surfer would more likely follow the out-links of the current page if the page has many out-links, assuming that it is a potential good hub page. Since the major computation is used to calculate the eigenvector \mathbf{v} , the time complexity of Dirichlet PageRank is the same as the standard PageRank.

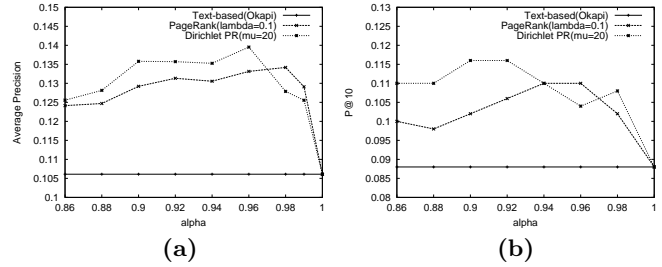
3. EXPERIMENTS

In this section, we compare the proposed Dirichlet PageRank algorithm with the standard PageRank as well as the text-only baseline.

As our data set, we used the ‘‘.GOV’’ test collection, which is an 18 gigabyte, 1.25 million documents 2002 partial crawl of the .gov domain used in TREC-2002 and TREC-2003 experiments for topic distillation. The queries for the ‘‘.GOV’’ data are fifty ‘‘topic distillation’’ topics created by NIST for TREC-2003 task with 10.32 relevant documents on average.

In our experiments, We used both the non-interpolated average precision (AvgPrec) and precision at 10 documents ($P@10$) as our evaluation metrics. We used the text-based Okapi retrieval method and the BM2500 weighting function as our baseline. The best parameters of the Okapi are set the same as paper [2], i.e. $k1 = 4.2, k3 = 1000, b = 0.8$. To test the effectiveness of different PageRanks, we chose the top 2000 documents according to Okapi scores and constructed two rankings of the documents: the ranking based on the Okapi scores and the PageRank ranking. The final ranking is obtained by combining these two rankings in a way similar to the one used in [2]:

$$\alpha \cdot \text{rank}_{\text{text}} + (1 - \alpha) \cdot \text{rank}_{\text{PageRank}}$$


Figure 1: Performance comparison along with α

We then rerank the 2000 documents according to the combined scores and select the top 1000 documents for evaluation. Apparently, the parameter α has impact on the performance. We vary α in the experiments and select the best results for comparison.

In Table 1, the best text-based result, the best PageRank result and the best Dirichlet PageRank result are listed. Both AvgPrec and $P@10$ are reported and compared. From the table, we can see that both PageRank algorithms improve the performance over the text-based method significantly. Furthermore, the Dirichlet PageRank achieves better performance than the standard PageRank on both AvgPrec (4.03% improvement) and $P@10$ (5.55% improvement). The Wilcoxon signed rank test indicates the AvgPrec improvement of Dirichlet PageRank over standard PageRank is statistically significant (p-value=0.034). This confirms that differentiating the pages with different number of out-links has the potential to emphasize the role of good hub pages. Note that our baseline result is not the same as the baseline in paper [2] because we use all the documents in ‘‘.GOV’’ data set while paper [2] only uses the documents with text/html format.

We also study the performance under different parameter settings. The results are displayed in Figure 1. In Figure 1(a), the average precision is plotted along with different α values. Dirichlet PageRank achieves the best result when $\alpha = 0.96$ and $\mu = 20$. The standard PageRank achieves the best result when $\alpha = 0.98$ and $\lambda = 0.1$. Figure 1(b) displays the comparison based on $P@10$. As can be seen from the figures, overall, the Dirichlet PageRank outperforms the standard PageRank.

4. CONCLUSIONS

In this poster, we introduce Dirichlet PageRank to make the jumping probability dynamically adapting the number of out-links of a page. Compared to the standard PageRank algorithm, the new algorithm favors following the out-links of potential good hub pages. Our experiments on the TREC data set confirms the effectiveness of Dirichlet PageRank.

5. REFERENCES

- [1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.
- [2] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma. Block-level link analysis. In *SIGIR*, pages 440–447, 2004.
- [3] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
- [4] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342, 2001.