The Effect of the *Back* Button in a Random Walk: Application for PageRank

Fabien Mathieu
Gyroweb – INRIA, LIRMM
34392 Montpellier Cedex 5 France
fmathieu@clipper.ens.fr

Mohamed Bouklit
LIRMM
34392 Montpellier Cedex 5 France
bouklit@lirmm.fr

ABSTRACT

Theoretical analysis of the Web graph is often used to improve the efficiency of search engines. The PageRank algorithm, proposed by [5], is used by the Google search engine [4] to improve the results of the queries.

The purpose of this article is to describe an enhanced version of the algorithm using a realistic model for the *back* button. We introduce a limited history stack model (you cannot click more than m times in a row), and show that when $m = 1$, the computation of this *Back* PageRank can be as fast as that of a standard PageRank.

Categories and Subject Descriptors

F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*Computations on matrices*

General Terms

Algorithms, Measurement

Keywords

Web analysis, PageRank, Random walk, flow, back button

1. INTRODUCTION

Since the introduction of the *PageRank* algorithm in 1998, numerous enhancement were made in both implementation and theorical efficiency. Using the stochastic aspect of the PageRank algorithm, the concept of *backoff process* was introduced by *Fagin et al.* [3] as an idealized model of browsing the web using both hyperlinks and the *back* button. This model allow the history stack to grow unboundedly. We introduce a bounded history stack, and show that in the special case of a one page history, there is an explicit and fast algorithm for computing the PageRank.

2. NOTATIONS

Let $G = (V, E)$ be a web graph, that is a set V of web pages linked to each other by a set E of edges.

If G is aperiodic and strongly connected, it is well known [6] that the iterative process

$$\forall v \in V, n \in \mathbb{N}, P_{n+1}(v) = \sum_{w \to v} \frac{P_n(w)}{d(w)}, \quad (1)$$

where $d(v)$ is the out-degree of $v \in V$, converges towards an unique probability P for any given probability P_0.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

However the web graph is far from being strongly connected [2]. One solution is to introduce a dumping factor d. The principle of the dumping factor is to "dump" the iterative process:

$$\forall v \in V, n \in \mathbb{N}, P_{n+1}(v) = d \sum_{w \to v} \frac{P_n(w)}{d(w)} + (1-d)S(v), \quad (2)$$

where S is a given probability on V[1].

A dumping factor is equivalent to working on a weighted strongly connected graph. If G is leafless, the limit P of (2) exists. Otherwise, normalization is needed.

3. *BACK* BUTTON MODEL

We suggest to refine the PageRank model by inserting the possibility to *return*. We choose a bounded history stack, so the PageRank algorithm is equivalent to a Markov chain with finite memory m. Potentially, this leads to consider all the possible paths in G of length m. For $m = 1$, this corresponds to the set E of the hyperlinks. We introduce two intuitive models for $m = 1$, one of them collapsing the working space from E to V. To begin with and for simplicity, we examine our *Back* button process without dumping.

3.1 Reversible *back*

In this model, we suppose that the web user can click at each state either on the links or on the *Back* button (the *Back* button is then considered as an outgoing link like the others). The probability of using the *Back* button is the same as that of using a given link. using *Back* button brings the user back to the previous state[2].

Let $P_n^{rb}(w, v)$ be the probability of being in v in the instant n coming from w in the instant $n-1$. $P_n^{rb}(w, v)$ is defined if $(w, v) \in E$ or $(v, w) \in E$. We can express the probability $P_n(v)$ of being in v at the instant n as follows:

$$P_n(v) = \sum_{w \leftrightarrow v} P_n^{rb}(w, v) \quad (3)$$

Note that because of the *Back* process, we work on the non-directed graph induced by G.

Working on the same principle, we deduce an equation expressing $P^{rb}(w, v)$: if $(w, v) \notin E$ (but (v, w) is), going from w to v implies using the *Back* button; then we were previously in w coming from v. Otherwise, either the *Back* button or the regular link can be used. Thus we have:

[1]Most of the time, $S \equiv \frac{1}{|V|}$, but some have suggested that it would be better to "personalize" it [1].

[2]Thus two consecutive uses of the *Back* button cancel each other.

$$P^{rb}_{n+1}(w,v) = \begin{cases} \frac{1}{d(w)+1}(P_n(w) + P^{rb}_n(v,w)) \text{ if } (w,v) \in E, \\ \frac{P^{rb}_n(v,w)}{d(w)+1} \text{ otherwise.} \end{cases} \quad (4)$$

Using (3) and (4) gives an iterative process for computing the new PageRank, but if $G' = (V, E')$ is the non-oriented graph induced by G, we have to use $|V| + |E'|$ variables instead of $|V|$ for the standard PageRank.

3.2 Irreversible *Back*

We now consider that the *Back* button cannot be used twice consecutively. This model, which seems more complex, as however three important advantages. First, it significantly reduces the stored PageRank by "greenhouse effect" in the end-nodes. Second, it is more appropriate to the insertion of a dumping factor (see 3.3). Finally it is less heavy on resource.

For $(w,v) \in E$, let $P^{ib}_n(w,v)$ be the probability to arrive at v using an hyperlink in w, and $\bar{P}^{ib}_n(v)$ the probability to arrive at v using the *Back* button. $\bar{P}^{ib}_{n+1}$ can be deduced from P^{ib}_n:

$$\bar{P}^{ib}_{n+1}(v) = \sum_{w \leftarrow v} \frac{P^{ib}_n(v,w)}{d(w)+1} \quad (5)$$

Then we can tell P^{ib}_{n+1} from P^{ib}_n and $\bar{P}^{ib}_n$:

$$P^{ib}_{n+1}(w,v) = \frac{1}{d(w)+1} \sum_{u \to w} P^{ib}_n(u,w) + \frac{\bar{P}^{ib}_n(w)}{d(w)} \quad (6)$$

We can note that $P^{ib}_{n+1}(w,v)$ does not depend on the arrival node v. We can then use P^{ib}_n on V instead of E, specifying only the departure node.

Equations (5) and (6) can now be written:

$$\bar{P}^{ib}_{n+1}(v) = P^{ib}_n(v) \sum_{w \leftarrow v} \frac{1}{d(w)+1} \quad (7)$$

$$P^{ib}_{n+1}(v) = \frac{1}{d(v)+1} \sum_{w \to v} P^{ib}_n(w) + \frac{\bar{P}^{ib}_n(v)}{d(v)} \quad (8)$$

3.3 *Back* button and dumping

For a real graph, insertion of the *Back* button ensures there is virtually no leaf, but the process may still not be irreducible, so we need to introduce a dumping factor. We made the choice to deactivate the *back* button after a crossing[3]. We can then merge (2), (7) and (8) to obtain:

$$\bar{P}^{ib}_{n+1}(v) = dP^{ib}_n(v) \left(\sum_{w \leftarrow v} \frac{1}{d(w)+1} \right) + (1-d)S(v) \quad (9)$$

$$P^{ib}_{n+1}(v) = d \left(\frac{1}{d(v)+1} \sum_{w \to v} P^{ib}_n(w) + \frac{\bar{P}^{ib}_n(v)}{d(v)} \right) \quad (10)$$

[3]Deactivating the *back* button after a crossing avoids to consider the $V \times V$ crossing transitions in the *Back* process.

4. EFFECTIVE COMPUTATION

4.1 Convergence

The process we made is stochastic (there is no blind way), aperiodic and irreducible (because of the dumping factor). The Perron-Frobenius theorem applies and ensures that the iterative process converges towards an unique fixed point.

4.2 Optimization

Using (9) and (10), we get an iterative way of calculating P^{ib}_n, and $P^{ib}_{n+1}(v)$ is equal to:

$$\sum_{w \to v} \frac{dP^{ib}_n(w)}{d(v)+1} + \sum_{w \leftarrow v} \frac{d^2 P^{ib}_{n-1}(v)}{d(v)(d(w)+1)} + \frac{d(1-d)S(v)}{d(v)} \quad (11)$$

Equation (11) is a two terms recurrence, but as we want to compute a fix point, the Gauss-Seidel method allows to use P^{ib}_n instead of P^{ib}_{n-1}; indeed one can approximate $P^{ib}_{n+1}(v)$ by:

$$\sum_{w \to v} \frac{dP^{ib}_n(w)}{d(v)+1} + \sum_{w \leftarrow v} \frac{d^2 P^{ib}_n(v)}{d(v)(d(w)+1)} + \frac{d(1-d)S(v)}{d(v)} \quad (12)$$

We remark that this iterative process has the same complexity that the standard PageRank computation.

Once P^{ib}_n has converged toward a vector P^{ib}, we obtain easily the asymptotic probability of presence P as follows:

$$P(v) = \sum_{w \to v} P^{ib}(w) + \bar{P}^{ib}(v) \quad (13)$$

5. CONCLUSION

We have proposed an alternative PageRank that can be obtained as easily that the standard PageRank and that should offer a better modelization of the web users. Computations made on a 8 millions pages graph showed that the top ranked pages differ from one model to another, yet both seemed interesting. We still have to merge this algorithm with a semantic pertinence-sort to be able to test this new model in the "real life".

6. REFERENCES

[1] S. Brin, R. Motwani, L. Page, and T. Winograd. What can you do with a Web in your Pocket? *Data Engineering Bulletin*, 21(2):37–47, 1998.

[2] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1-6):309–320, 2000.

[3] R. Fagin, A. R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons" (extended abstract). pages 484–493, 2000.

[4] Google. http://www.google.com/, 1998.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Computer Science Department, Stanford University, 1998.

[6] L. Saloff-Coste. Lectures on finite Markov chains. In G. G. E. Giné and L. Saloff-Coste, editors, *Lecture Notes on Probability Theory and Statistics*, number 1665 in LNM, pages 301–413. Springer Verlag, 1996.