PageRank algorithm and Monte Carlo methods in PageRank Computation

Özlem Salehi
2007103004

Abstract

PageRank is the algorithm used by the Google search engine for ranking web pages. PageRank Algorithm calculates for each page a relative importance score which can be interpreted as the frequency of how often a page is visited by a surfer. The purpose of this work is to provide a mathematical analysis of the PageRank Algorithm. We analyze the random surfer model and the linear algebra behind it which complements the discussion of Markov Chains in matrix algebra. We also study Monte Carlo type methods for PageRank computation, which have several advantages over the Power method used by Google: Monte Carlo methods provide good estimation of the PageRank for relatively important pages already after one iteration; Monte Carlo methods have natural parallel implementation; and finally, Monte Carlo methods allow to perform continuous update of the PageRank as the structure of the Web changes.

1 Introduction

The World Wide Web contains an infinite resource of information. Without the help of search engines, it would be impossible for us to find the necessary information among billions of web pages. A search engine is an information retrieval system for the web, which when queried on some topic returns a list of documents which share same content or relationship to the topic. A web search engine possesses three major components: a *crawler* which collects and stores large amounts of raw data from the web, an *indexer* which extracts information from the data collected by the crawler and builds an index so that the data can be searched efficiently for relevant keywords and phrases and a *query engine* which responds in real time to queries from users. An important component of the query engine is the *ranking algorithm*, which attempts to rank web pages in order of their relevance to the query.

The first search engine was Archie, which was created in 1990, and was basically a database of web filenames which could be queried by users. The first commercial search engines came out in 1994, with the introduction of engines such as Lycos, Infoseek, and AltaVista. These search engines were text-based, and in that they ranked pages higher which had a high frequency of the query phase. Such engines were susceptible to spamming, where words are repeated in a document to increase its ranking [12].

Rapid growth of World Wide Web required more effective ranking algorithms. The idea of using the link structure of the Web first arose in 1990's. In 1997 HITS algorithm proposed by Jon Kleinberg appeared in [1]. In 1998, PageRank algorithm developed by Larry Page and Sergey Brin appeared in [2]. A third algorithm, Stochastic Approach for Link Structure Analysis (SALSA) which combined ideas from PageRank and HITS was proposed by Ronny Lempel and Shlomo Moran in 2000 [4].

Our focus in this paper is the PageRank algorithm, the algorithm used by the Google search engine. PageRank uses the link structure of the Web to produce a relative importance score for every page so that more important pages can be presented first when a user searches the web. The motivation behind the algorithm is the citation analysis techniques which date back to 1940's. The main goal was to rank academic documents by counting the number of times they have been cited. One can think of every web page as an academic document and every link as an academic citation.

In order to keep up with constant modifications in the World Wide Web, Google updates its PageRank values at least once per month. Google uses the power method in PageRank calculation which will be discussed in Section 3. Several proposals have been put forward to accelerate the power method in [6, 7, 8, 9].

Here in contrast, we study Monte Carlo type methods in computation of PageRank. Monte Carlo methods provide approximate solutions to mathematical problems by performing repeated sampling experiments. Previously, Monte Carlo type methods have been applied to PageRank in [9, 5]. Probabilistic Monte Carlo type methods have several advantages over deterministic methods: the PageRank of important pages is determined with high accuracy already after the first iteration; MC methods have natural parallel implementation; and MC methods allow continuous update of the PageRank as the structure of the Web changes.

The paper is designed as follows. In section 2, mathematical formulation of PageRank is presented. In Section 3, power method which is used by Google to compute PageRank is analyzed. In Section 4, Monte Carlo methods to compute PageRank is proposed and they are analyzed in Section 5. We show that the PageRank of relatively important pages can be determined with high accuracy even after the first iteration.
We denote the indicator function by $\mathbf{1}$.

2 Formulation of PageRank

The main idea behind assigning a score to any given web page is that the page's score is derived from the links made to that page. Let u be a web page. Let B_u be the set of pages that points to u and let N_u be the number of outgoing links from u. We can define a simple ranking R as follows[10]:

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{N_v}. \tag{1}$$

Note that in this formulation, a link to page u from an important page increases page u's score more than a link from an unimportant page. Also, the rank is divided by the number of outgoing links, since the probability of following the link to page u decreases as the number of outgoing links increases.

In matrix notation, let Q be a square matrix with rows and columns corresponding to web pages with:

$$Q_{ij} = \begin{cases} 1/N_i & \text{if page i links to page j,} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{r}$ be the PageRank vector with entries $R(u)$, $u = 1...n$ where n is the number of pages. Then we have $\boldsymbol{r}Q = \boldsymbol{r}$ so that $\boldsymbol{r}$ is a left eigenvector of Q with eigenvalue 1.

Random Surfer Model The definition of PageRank above is simply the stationary distribution of a random walk on the graph of web. We can think of a random surfer which clicks with equal probability one of the outgoing links and moves to another page. The definition has some shortcomings which will be discussed next.

Definition 1. *A square matrix is called a* ***row stochastic matrix*** *if all of its entries are nonnegative and entries in each row sum to one.*

Dangling Nodes A web with dangling nodes (pages with no outgoing links) produces a matrix Q which contains one or more rows of all zeroes. Such a matrix must have all eigenvalues less than or equal to 1, but 1 need not be an eigenvalue for Q. To be able to use some properties of row stochastic matrices we eliminate the dangling pages. Let P be the matrix defined as:

$$P_{ij} = \begin{cases} 1/N_i & \text{if page i links to page j,} \\ 1/n & \text{if page i has no outgoing links,} \\ 0 & \text{otherwise,} \end{cases}$$

where n is the number of pages. Now we have a random surfer which jumps to a random page when stuck in a dangling node.

Proposition 1. *Every row stochastic matrix has 1 as an eigenvalue.*

Proof. Let M be an $n \times n$ row stochastic matrix and let $\underline{\mathbf{1}}$ denote an n dimensional row vector with all entries equal to 1. Since M is row stochastic it is easy to see that for all $i = 1...n$, $(M\underline{1}^T)_i = \sum_{j=1}^n M_{ij}(\underline{1}^T)_j = \sum_{j=1}^n M_{ij}1 = 1 = (\underline{1}^T)_i$ so that 1 is an eigenvalue for P. □

Non-Unique Rankings Let $V_1(P)$ denote the eigenspace for eigenvalue 1 of matrix P. For our rankings it is desirable that the dimension of $V_1(P)$ is equal to one so that there is a unique eigenvector $\boldsymbol{r}$ that we can use for importance scores. However it is not always the case. If a web W consists of k disconnected subsets, then $dim(V_1(P)) \geq k$ and hence there is no unique importance score vector. This makes intuitive sense since one would expect difficulty in finding a common reference frame comparing the scores of pages in one subweb with those in another subweb [10].

Modification to the matrix P We assume that a random surfer continues following the links with probability c or jumps to a random page with probability $1 - c$. By this modification, we get a connected web and we solve the problem mentioned above. Now, we can define the Google matrix as follows:

$$\tilde{P} = cP + (1 - c)\frac{1}{n}E, \tag{2}$$

where E is a $n \times n$ matrix of all ones and the value of c is chosen to be 0.85 in [2].

Proposition 2. *If M is positive and row stochastic, then any eigenvector in $V_1(M)$ has all positive or all negative components.*

Proof. Suppose that $\boldsymbol{x} \in V_1(M)$ contains elements of mixed sign. From $\boldsymbol{x}M = \boldsymbol{x}$, we have $\sum_{i=1}^n x_i M_{ij} = x_j$ and the summands $x_i M_{ij}$ are of mixed sign. As a result we have

$$|x_j| = \left|\sum_{i=1}^n x_i M_{ij}\right| < \sum_{i=1}^n |x_i| M_{ij}$$

Summing both equalities from $j = 1$ to $j = n$ and swapping the i and j summation we find

$$\sum_{j=1}^{n} |x_j| < \sum_{j=1}^{n} \sum_{i=1}^{n} |x_i| M_{ij} = \sum_{i=1}^{n} |x_i| \sum_{j=1}^{n} M_{ij} = \sum_{i=1}^{n} |x_i|,$$

a contradiction. Hence $\boldsymbol{x}$ cannot contain both positive and negative elements. □

Proposition 3. *Let $\boldsymbol{v}$ and $\boldsymbol{w}$ be linearly independent vectors in $\mathbb{R}^m$, $m \geq 2$. Then for some values of s and t that are not both zero, the vector $\boldsymbol{x} = s\boldsymbol{v} + t\boldsymbol{w}$ has both positive and negative components.*

Proof. Linear independence implies neither $\boldsymbol{v}$ nor $\boldsymbol{w}$ is zero. Let $d = \sum_j v_j$. If $d = 0$ then $\boldsymbol{v}$ must contain components of both sign, and taking $s = 1$ and $t = 0$ yields the conclusion. If $d \neq 0$ set $s = -\frac{\sum_j w_j}{d}$, $t = 1$, and $\boldsymbol{x} = s\boldsymbol{v} + t\boldsymbol{w}$. Since $\boldsymbol{v}$ and $\boldsymbol{w}$ are independent, $\boldsymbol{x} \neq 0$. However, $\sum_j x_j = 0$. We conclude that $\boldsymbol{x}$ has both positive and negative components. □

Lemma 1. *If M is positive and row stochastic then $V_1(M)$ has dimension 1.*

Proof. Suppose there are two linearly independent vectors $\boldsymbol{v}$ and $\boldsymbol{w} \in V_1(M)$. Then for any real number s and t that are not both zero, the nonzero vector $\boldsymbol{x} = s\boldsymbol{v} + t\boldsymbol{w}$ must be in $V_1(M)$ and so have components that are all positive or all negative by Proposition 2. But by Proposition 3, for some choice of s and t the vector $\boldsymbol{x}$ contains components of mixed sign and we obtain a contradiction. Hence, we conclude that $V_1(M)$ has dimension 1. □

By the above lemma, we can conclude that $V_1(\tilde{P})$ is one dimensional and the corresponding eigenvector has entirely positive or negative components. We are thus guaranteed the existence of a unique eigenvector $\boldsymbol{\pi} \in V_1(\tilde{P})$ with positive components such that

$$\boldsymbol{\pi}\tilde{P} = \boldsymbol{\pi}, \quad \sum_i \pi_i = 1. \tag{3}$$

3 Power Method in PageRank Computation

The method used by Google in computation of PageRank is the power method [2]. The power method is that one starts with an initial vector $\boldsymbol{x^{(0)}}$, generates the sequence $\boldsymbol{x^{(k)}} = \boldsymbol{x^{(k-1)}}M$ and then lets k approach infinity. The vector $\boldsymbol{x^{(k)}}$ is an approximation to the eigenvector for the largest eigenvalue of M. We first give a proposition and then state a theorem which proves that the PageRank vector can be computed as the limit of iterations of the Power method.

Definition 2. *The 1-norm of a vector $\boldsymbol{v}$ is $||\boldsymbol{v}|| = \sum_j |v_j|$.*

Proposition 4. *Let M be a positive row stochastic $n \times n$ matrix and let V denote the subspace of $\mathbb{R}^n$ consisting of vectors $\boldsymbol{v}$ such that $\sum_j v_j = 0$. Then $\boldsymbol{v}M \in V$ for any $\boldsymbol{v} \in V$, and $||\boldsymbol{v}M|| \leq c||\boldsymbol{v}||$ for any $\boldsymbol{v} \in V$, where $c = max_{1 \leq i \leq n}|1 - 2min_{1 \leq j \leq n} M_{ij}| < 1$.*

Proof. Let $\boldsymbol{w} = \boldsymbol{v}M$, so that $w_j = \sum_{i=1}^{n} v_i M_{ij}$ and

$$\sum_{j=1}^{n} w_j = \sum_{j=1}^{n} \sum_{i=1}^{n} v_i M_{ij} = \sum_{i=1}^{n} v_i \sum_{j=1}^{n} M_{ij} = \sum_{i=1}^{n} v_i = 0.$$

Hence $\boldsymbol{w} = \boldsymbol{v}M \in V$. To prove the bound in the proposition note that

$$||\boldsymbol{w}|| = \sum_{j=1}^{n} e_j w_j = \sum_{j=1}^{n} e_j \sum_{i=1}^{n} v_i M_{ij} = \sum_{i=1}^{n} v_i \sum_{j=1}^{n} e_j M_{ij} = \sum_{i=1}^{n} a_i v_i,$$

where $e_j = sgn(w_j)$ and $a_i = \sum_{j=1}^{n} e_j M_{ij}$. Note that the e_j are not all of one sign, since $\sum_j w_j = 0$. Then it is easy to see that

$$-1 < -1 + 2 \min_{1 \leq j \leq n} M_{ij} \leq a_i \leq 1 - 2 \min_{1 \leq j \leq n} M_{ij} < 1$$

We can thus bound $|a_i| \leq |1 - 2min_{1 \leq i \leq n} M_{ij}| < 1$. Observe that $c < 1$ and $|a_i| \leq c$ for all i. Then we have

$$||\boldsymbol{w}|| = \sum_{i=1}^{n} a_i v_i = \left| \sum_{i=1}^{n} a_i v_i \right| \leq \sum_{i=1}^{n} |a_i||v_i| \leq c \sum_{i=1}^{n} |v_i| = c||\boldsymbol{v}||$$

which proves the proposition. □

Theorem 1. *Every positive row stochastic matrix M has a unique vector $\boldsymbol{q}$ with positive components such that $\boldsymbol{q}M = \boldsymbol{q}$ with $||\boldsymbol{q}|| = 1$. The vector $\boldsymbol{q}$ can be computed as $\boldsymbol{q} = \lim_{k \to \infty} \boldsymbol{x}^{(0)} M^k$ for any initial guess $\boldsymbol{x}^{(0)}$ with positive components such that $||\boldsymbol{x}^{(0)}|| = 1$.*

Proof. By the above propositions it is clear that there is a unique vector $\boldsymbol{q}$ with positive components such that $\boldsymbol{q}M = \boldsymbol{q}$ with $||\boldsymbol{q}|| = 1$. Let $\boldsymbol{x}^{(0)}$ be any vector in $\mathbb{R}^n$ with positive components such that $||\boldsymbol{x}^{(0)}|| = 1$. We can write $\boldsymbol{x}^{(0)} = \boldsymbol{q} + \boldsymbol{v}$ where $\boldsymbol{q} \in V$ (V as in Proposition 4). We find that $\boldsymbol{x}^{(0)} M^k = \boldsymbol{q} M^k + \boldsymbol{v} M^k = \boldsymbol{q} + \boldsymbol{v} M^k$. As a result

$$\boldsymbol{x}^{(0)} M^k - \boldsymbol{q} = \boldsymbol{v} M^k.$$

By Proposition 4 and doing simple induction, $||\boldsymbol{v}M^k|| \leq c^k ||\boldsymbol{v}||$ for $0 \leq c < 1$ (c as in Proposition 4) and so $\lim_{k \to \infty} ||\boldsymbol{v}M^k|| = 0$. We conclude that $\lim_{k \to \infty} \boldsymbol{x}^{(0)} M^k = \boldsymbol{q}$. □

We have thus proved that the vector $\boldsymbol{\pi}$ may be computed as the limit of iterations $\boldsymbol{\pi}^{(\boldsymbol{k})} = \boldsymbol{\pi}^{(\boldsymbol{k-1})} \tilde{P}$.

The Power method stops when the required precision ε is achieved. The number of flops needed for the method to converge is of the order $\frac{log \varepsilon}{log c} nnz(\tilde{P})$, where $nnz(\tilde{P})$ is the number of non-zero elements of the matrix $nnz(\tilde{P})$ [9]. We note that the relative error decreases uniformly for all pages.

4 Monte Carlo Methods in PageRank Computation

Probabilistic Monte Carlo (MC) methods have some principle advantages over the deterministic methods: the PageRank of important pages is determined with high accuracy already after the first iteration; MC methods have natural parallel implementation; and MC methods allow continuous update of the PageRank as the structure of the Web changes [11]. From equations (2) and (3) we get,

$$\boldsymbol{\pi} = \boldsymbol{\pi} \tilde{P} = \boldsymbol{\pi} c P + \frac{1-c}{n} \boldsymbol{\pi} E = \boldsymbol{\pi} c P + \frac{1-c}{n} \underline{\mathbf{1}}$$

$$\boldsymbol{\pi}(I - cP) = \frac{1-c}{n} \underline{\mathbf{1}}.$$

Monte Carlo algorithms are motivated by the following formula

$$\boldsymbol{\pi} = \frac{1-c}{n}\underline{\mathbf{1}}[I-cP]^{-1}, \tag{4}$$

which directly follows from above, where $\underline{\mathbf{1}}$ is the $1 \times n$ row vector of ones. Since $[I-cP]$ is non singular[3], $\boldsymbol{\pi}$ is uniquely defined by equation (4). Now we can write PageRank of page j as

$$\pi_j = \frac{1-c}{n}\sum_{i=1}^{n}[I-cP]^{-1}_{ij}. \tag{5}$$

Consider a random walk $\{X_t\}_{t\geq 0}$ that starts from a randomly chosen page. Assume that at each step the random walk makes a transition according to the matrix P with probability c or terminates with probability $(1-c)$. Then it follows from (4) that the end-point of this random walk (the last visited page before the random walk terminates) appears to be a sample from the distribution $\boldsymbol{\pi}$. Thus, after repeating the process many times, the estimate for π_j for $j = 1...n$ can be determined as the number of times the random walk terminates at page j divided by the total number of random walks. Hence, one can suggest the following algorithm:

Algorithm 1. ***MC end-point with random start.*** *Simulate N runs of the random walk $\{X_t\}_{t\geq 0}$ initiated at a randomly chosen page. Evaluate π_j as a fraction of N random walks which end at page j.*

Probability that k out of N random walks end at page j can be given by $\binom{N}{k}\pi_j^k(1-\pi_j)^{N-k}$. Let $\hat{\pi}_{j,N}$ be the estimator of π_j obtained by Algorithm 1. Let K_j be the number of random walks that end at page j. Then,

$$\mathbb{E}(\hat{\pi}_{j,N}) = \mathbb{E}\left[\frac{K_j}{N}\right] = \frac{\mathbb{E}[K_j]}{N} = \frac{N\pi_j}{N} = \pi_j,$$

$$Var(\hat{\pi}_{j,N}) = Var\left(\frac{K_j}{N}\right) = \frac{1}{N^2}Var(K_j) = \frac{1}{N^2}N\pi_j(1-\pi_j) = N^{-1}\pi_j(1-\pi_j).$$

In order to improve the estimator $\hat{\pi}_j$, one can think of various ways of variance reduction. We can view π_j as a given number $(1/n)$ multiplied by a sum of conditional probabilities $v_{ij} = (1-c)[I-cP]^{-1}_{ij}$ that the random walk ends at j given that it started at i. Since n is known, an unnecessary randomness in experiments can be avoided by taking $N = nm$ and initiating the random walk exactly m times from each page in a cyclic fashion, rather than jumping N times to a random page. This results in the following algorithm.

Algorithm 2. ***MC end-point with cyclic start.*** *Simulate $N = nm$ runs of random walk $\{X_t\}_{t\geq 0}$ initiated at each page exactly m times. Evaluate π_j as a fraction of N random walks which end at page j.*

Let K_{ij} be the number of random walks initiated at i and ended at j. Then the estimator suggested by Algorithm 2 can be expressed as $\hat{\hat{\pi}}_j = \frac{1}{N}\sum_{i=1}^{n} K_{ij}$.

For this estimator, we have

$$\mathbb{E}[\hat{\hat{\pi}}_j] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{n}K_{ij}\right] = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n}\mathbb{E}[K_{ij}] = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n}mv_{ij} = \frac{1}{n}\sum_{i=1}^{n}v_{ij} = \pi_j$$

$$
\begin{aligned}
Var(\hat{\hat{\pi}}_j) &= Var\left(\frac{1}{N}\sum_{i=1}^{n} K_{ij}\right) = \frac{1}{N^2}\sum_{i=1}^{n} Var(K_{ij}) = \frac{1}{N^2}\sum_{i=1}^{n} m(1-v_{ij})(v_{ij}) \\
&= \frac{m}{N^2}\left(\sum_{i=1}^{n} v_{ij} - \sum_{i=1}^{n} v_{ij}^2\right) = \frac{m}{N^2}\left(n\pi_j - \sum_{i=1}^{n} v_{ij}^2\right) \\
&= N^{-1}\left(\pi_j - \frac{1}{n}\sum_{i=1}^{n} v_{ij}^2\right)
\end{aligned}
$$

By using Cauchy-Schwarz inequality we get $\left(\sum_{i=1}^{n}\frac{1}{n}v_{ij}\right)^2 < \sum_{i=1}^{n}\frac{1}{n^2}\sum_{i=1}^{n} v_{ij}^2$. Using this inequality we obtain

$$
Var(\hat{\hat{\pi}}_j) < N^{-1}\left(\pi_j - \frac{1}{n^2}\left(\sum_{i=1}^{n} v_{ij}\right)^2\right) = N^{-1}(\pi_j - \pi_j^2) = N^{-1}\pi_j(1-\pi_j) = Var(\hat{\pi}_j).
$$

Besides variance reduction, the estimator $Var(\hat{\hat{\pi}}_j)$ has important advantages in implementation because it avoids the difficulty of picking a page at random.

Another way of reducing the variance is to rewrite the formula (4) as

$$
\boldsymbol{\pi} = \frac{1-c}{n}\underline{\mathbf{1}}\sum_{k=0}^{\infty} c^k P^k. \tag{6}
$$

Hence we get,

$$
\pi_j = \frac{(1-c)}{n}\sum_{i=1}^{n}\sum_{k=0}^{\infty} c^k P_{ij}^k. \tag{7}
$$

Note that $\sum_{k=0}^{\infty} c^k P_{ij}^k$ can be regarded as the average number of times that the random walk $\{X_t\}_{t\geq 0}$ visits a page j given that the random walk started at page i. Thus, we can propose an estimator based on a complete path of the random walk.

Algorithm 3. ***MC complete path.*** *Simulate the random walk $\{X_t\}_{t\geq 0}$ exactly m times from each page. Evaluate π_j as the total number of visits to page j multiplied by $(1-c)/(nm)$.*

Now let $\mathcal{K}_{ij}$ denote the number of visits to page j given that the random walk $\{X_t\}_{t\geq 0}$ initiated from page i. We can express the estimator for Algorithm 3 as $\tilde{\pi}_j = \frac{1-c}{nm}\sum_{i=1}^{n}\mathcal{K}_{ij}$. We denote the indicator function by $\mathbf{1}$. Then we have

$$
\begin{aligned}
\mathbb{E}[\tilde{\pi}_j] &= \mathbb{E}\left[\frac{1-c}{nm} m \sum_{i=1}^{n}\mathcal{K}_{ij}\right] = \frac{1-c}{n}\sum_{i=1}^{n}\mathbb{E}[\mathcal{K}_{ij}] = \frac{1-c}{n}\sum_{i=1}^{n}\sum_{k=0}^{\infty}\mathbb{E}[\mathbf{1}_{\{X_k=j\}}|X_0=i] \\
&= \frac{1-c}{n}\sum_{i=1}^{n}\sum_{k=0}^{\infty}\mathbb{P}(X_k=j|X_0=i) = \frac{1-c}{n}\sum_{i=1}^{n}\sum_{k=0}^{\infty} c^k P_{ij}^k \\
&= \pi_j.
\end{aligned}
$$

Algorithm 3 can be further improved by getting rid of the artifacts in matrix P related to dangling pages. When a random walk reaches a dangling node, it jumps with uniform probability to an arbitrary page. Clearly, it is more efficient to terminate the random walk once it reaches a dangling node. Thus, we aim to rewrite (4) in terms of the original hyperlink matrix Q defined as

$$Q_{ij} = \begin{cases} 1/N_i & \text{if page i links to page j,} \\ 0 & \text{otherwise.} \end{cases}$$

Denote by $\mathcal{I}_0$ the set of dangling pages and by $\mathcal{I}_1 = \{1...n\}\backslash\mathcal{I}_0$ the set of pages which have at least one outgoing link. It follows from (2) and (3)

$$\pi_j = c\sum_{i=1}^{n} \pi_i P_{ij} + \frac{(1-c)}{n}\sum_{i=1}^{n}\pi_i = c\sum_{i=1}^{n}\pi_i Q_{ij} + \gamma, \tag{8}$$

$$\gamma = \frac{c}{n}\sum_{i\in\mathcal{I}_0}\pi_i + \frac{(1-c)}{n} < \frac{1}{n}. \tag{9}$$

Writing equation (8) as $\boldsymbol{\pi} = \boldsymbol{\pi} cQ + \gamma\underline{\mathbf{1}}$, leads to the new expressions for $\boldsymbol{\pi}$ and π_j:

$$\boldsymbol{\pi} = \gamma\underline{\mathbf{1}}[I - cQ]^{-1}, \tag{10}$$

$$\pi_j = \gamma\sum_{i=1}^{n}[I - cQ]^{-1}_{ij}. \tag{11}$$

Consider now a random walk $\{Y_t\}_{t\geq 0}$ which follows links exactly as $\{X_t\}_{t\geq 0}$ except the transitions are governed by matrix Q instead of matrix P. Thus, the random walk $\{Y_t\}_{t\geq 0}$ can be terminated at each step with probability $(1-c)$ or when it reaches a dangling page. The element w_{ij} of the matrix $W = [I - cQ]^{-1}$ is the average number of visits of $\{Y_t\}_{t\geq 0}$ to page j given that the random walk started at page i. Denote $w_{.j} = \sum_{i=1}^{n} w_{ij}$. Summing π_j in equation (11) from $j = 1$ to $j = n$ we get $\sum_{j=1}^{n}\pi_j = \gamma\sum_{j=1}^{n}\sum_{i=1}^{n} w_{ij}$. Since the coordinates of $\boldsymbol{\pi}$ sum up to 1, we have $1 = \gamma\sum_{j=1}^{n} w_{.j}$. Then it follows,

$$\gamma = \left(\sum_{j=1}^{n} w_{.j}\right)^{-1}. \tag{12}$$

Replacing γ back in equation (11) we obtain

$$\pi_j = \left(\sum_{j=1}^{n} w_{.j}\right)^{-1} w_{.j}. \tag{13}$$

This calls for another version of the complete path method.

Algorithm 4. ***MC complete path stopping at dangling nodes.*** *Simulate the random walk $\{Y_t\}_{t\geq 0}$ starting exactly m times from each page. Evaluate π_j as the total number of visits to page j divided by the total number of visited pages.*

Let $\mathcal{W}_{ij}$ be a random variable distributed as number of visits to page j by the random walk $\{Y_t\}_{t\geq 0}$ given that the random walk initiated at page i. Let $\mathcal{W}_{ij}^{(l)}$, $l \geq 1$ be independent random variables distributes as $\mathcal{W}_{ij}$. Then the estimator produced by Algorithm 4 can be written as

$$\bar{\pi}_j = \left(\sum_{l=1}^{m}\sum_{j=1}^{n}\mathcal{W}_{.j}^{(l)}\right)^{-1}\left(\sum_{l=1}^{m}\mathcal{W}_{.j}^{(l)}\right) \tag{14}$$

Analysis of this estimator will be presented in the next section. We note that the complete path versions of Monte Carlo methods also admit a random start. The corresponding algorithm is as follows.

Algorithm 5. ***MC complete path with random start stopping at dangling nodes.*** *Simulate N samples of the random walk $\{Y_t\}_{t\geq 0}$ started at a random page. Evaluate π_j as the total number of visits to page j divided by the total number of visited pages.*

One can show that Algorithm 4 provides an estimator with a smaller variance than Algorithm 5. Let $\mathcal{W}_{uj}$ be the number of visits to page j from a randomly chosen page $u \in \{1...n\}$. Then, we have

$$\begin{aligned}
Var(\mathcal{W}_{uj}) &= Var(\sum_{i=1}^{n} \mathbf{1}_{\{u=i\}}\mathcal{W}_{ij}) \\
&= \sum_{i=1}^{n} Var(\mathbf{1}_{\{u=i\}}\mathcal{W}_{ij}) + 2\sum_{i<k}^{n} Cov(\mathbf{1}_{\{u=i\}}\mathcal{W}_{ij}, \mathbf{1}_{\{u=k\}}\mathcal{W}_{kj}) \\
&= \sum_{i=1}^{n} \mathbb{E}[(\mathbf{1}_{\{u=i\}}\mathcal{W}_{ij})^2] - \sum_{i=1}^{n} \mathbb{E}[\mathbf{1}_{\{u=i\}}\mathcal{W}_{ij}]^2 + 2\sum_{i<k}^{n} \mathbb{E}[\mathbf{1}_{\{u=i\}}\mathcal{W}_{ij}\mathbf{1}_{\{u=k\}}\mathcal{W}_{kj}] \\
&- 2\sum_{i<k}^{n} \mathbb{E}[\mathbf{1}_{\{u=i\}}\mathcal{W}_{ij}]\mathbb{E}[\mathbf{1}_{\{u=k\}}\mathcal{W}_{kj}] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}^2] - \frac{1}{n^2}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}]^2 + 0 - \Big(\Big(\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}]\Big)^2 - \frac{1}{n^2}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}]^2\Big) \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}^2] - \Big(\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}]\Big)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} Var(\mathcal{W}_{ij}) + \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}^2] - \Big(\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}]\Big)^2 \\
&> \frac{1}{n}\sum_{i=1}^{n} Var(\mathcal{W}_{ij}).
\end{aligned}$$

since $\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}^2] > \Big(\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathcal{W}_{ij}]\Big)^2$ follows from the Cauchy-Schwarz inequality.

Note that Algorithm 4 generates m samples of the sum $\sum_{i=1}^{n}(\mathcal{W}_{ij})$, whereas Algorithm 5 generates $N = mn$ samples of $\mathcal{W}_{uj}$. Hence, Algorithm 4 generates random variables with smaller variance in both numerator and denominator of (14).

5 Analysis of Monte Carlo methods

From the preliminary analysis of the previous section, we can already conclude that MC algorithms with cyclic start are preferable to the analogous MC algorithms with random start. In the present section we analyze and compare MC complete path stopping at dangling nodes with MC end point. We show that MC complete path stopping at dangling nodes outperforms MC-end-point.

We start by studying the properties of $\mathcal{W}_{ij}$'s. Denote by q_{ij} the probability that starting from page i, the random walk $\{Y_t\}_{t\geq 0}$ reaches page j. Note that in this definition, $q_{jj} < 1$ is the probability to return to page j given that the random walk started at page j. It follows that $\mathcal{W}_{ij}$ has geometric distribution with parameter $1 - q_{jj} \geq 1 - c$:

$$\begin{aligned}
&\mathbb{P}(\mathcal{W}_{jj} = k) = q_{jj}^{k-1}(1 - q_{jj}),\\
&\mathbb{E}[\mathcal{W}_{jj}] = \frac{1}{1 - q_{jj}},\\
&Var(\mathcal{W}_{jj}) = \frac{q_{jj}}{(1 - q_{jj})^2}.
\end{aligned}$$

Further, one can show that

$$\begin{aligned}
&\mathbb{P}(\mathcal{W}_{ij} = k) = \begin{cases} 1 - q_{ij}, & x = 0 \\ q_{ij}\mathbb{P}(\mathcal{W}_{ij} = k), & x = 1, 2... \end{cases}\\
&\mathbb{E}[\mathcal{W}_{ij}] = w_{ij} = q_{ij}\mathbb{E}[\mathcal{W}_{ij}] = \frac{q_{ij}}{1 - q_{ij}}, \qquad (15)
\end{aligned}$$

$$Var(\mathcal{W}_{ij}) = \mathbb{E}[\mathcal{W}_{ij}^2] - \mathbb{E}[\mathcal{W}_{ij}]^2 = q_{ij}\mathbb{E}[\mathcal{W}_{jj}^2] - w_{ij}^2 = q_{ij}\frac{1 + q_{jj}}{(1 - q_{jj})^2} - w_{ij}^2 = \frac{1 + q_{jj}}{1 - q_{jj}} w_{ij} - w_{ij}^2.$$

Assuming that all $\mathcal{W}_{.j}$'s are independent, we immediately obtain

$$\begin{aligned}
&\mathbb{E}[\mathcal{W}_{.j}] = \sum_{i=1}^{n} w_{ij} = w_{.j}, \qquad (16)\\
&Var(\mathcal{W}_{.j}) = \frac{1 + q_{jj}}{1 - q_{jj}} w_{.j} - \sum_{i=1}^{n} w_{ij}^2 < \frac{1 + q_{jj}}{1 - q_{jj}} w_{.j}, \qquad (17)\\
&\mathbb{E}[W] = \sum_{j=1}^{n} w_{.j} = \gamma^{-1}.
\end{aligned}$$

Let the empirical mean $\bar{\mathcal{W}}_{ij} = \frac{1}{m}\sum_{l=1}^{m} \mathcal{W}_{ij}^{(l)}$ be the estimator of w_{ij}, and view $\bar{\mathcal{W}}_{.j} = \sum_{j=1}^{n} \bar{\mathcal{W}}_{ij}$ and $\bar{\mathcal{W}} = \sum_{j=1}^{n} \bar{\mathcal{W}}_{.j}$ as estimators of $w_{.j}$ and γ^{-1} respectively. The estimator (14) can be written as

$$\bar{\pi}_j = \bar{\mathcal{W}}^{-1}\bar{\mathcal{W}}_{.j}. \qquad (18)$$

Since the first term in (18) is same for every j, the estimator $\bar{\pi}_j$ is completely determined by $\bar{\mathcal{W}}_{.j}$.

The following lemma gives us upper bounds for the expected value and the variance of number of transitions of the random walk $\{Y_t\}_{t\geq 0}$, which will be useful in our further discussion.

Lemma 2. *Let $\mathcal{W}_{i.} = \sum_{j=1}^{n} \mathcal{W}_{ij}$ be the length of random walk $\{Y_t\}_{t\geq 0}$ initiated at page i. Then for all dangling nodes $i \in \mathcal{I}_0$, it holds $\mathcal{W}_{i.} = 1$ and for non-dangling nodes $i \in \mathcal{I}_1$, $\mathbb{E}[\mathcal{W}_{i.}] \leq \frac{1}{1-c}$ and $Var(\mathcal{W}_{i.}) \leq \frac{c(1+c^3)}{(1-c)^2}$.*

Proof. The statement for dangling nodes is obvious. For non dangling nodes, the statement of the lemma follows from the fact $\mathcal{W}_{i.} \overset{d}{=} min\{X, N_i\}$, where N_i is the number of transitions needed to reach a dangling node from page i, and X has geometric distribution with parameter

$1-c$ with $\mathbb{E}[X] = \frac{1}{1-c}$ and $Var(X) = \frac{c}{(1-c)^2}$. The upper bound for the expectation of $\mathcal{W}_{i.}$ follows directly from the identity $\mathcal{W}_{i.} \overset{d}{=} min\{X, N_i\}$.

For the variance, by using the law of total variance we write

$$Var(\mathcal{W}_{i.}) = \mathbb{E}[Var(\mathcal{W}_{i.}|N_i)] + Var(\mathbb{E}[\mathcal{W}_{i.}|N_i]).$$

For the first term of the equality, our claim is that $\mathbb{E}[Var(\mathcal{W}_{i.}|N_i)] < Var(X)$. Conditioning on events $N_i = x$,

$$Var(\mathcal{W}_{i.}|N_i = x) - Var(X) = \mathbb{E}[\mathcal{W}_{i.}^2|N_i = x] - \mathbb{E}[X^2] + \mathbb{E}[X]^2 - \mathbb{E}[\mathcal{W}_{i.}|N_i = x]^2.$$

First, we look at the first two terms of the above equality.

$$\begin{aligned}
&\mathbb{E}[\mathcal{W}_{i.}^2|N_i = x] - \mathbb{E}[X^2] = \sum_{k=1}^{x-1} k^2 c^{k-1}(1-c) + x^2\mathbb{P}(X \geq x) - \sum_{k=1}^{\infty} k^2 c^{k-1}(1-c)\\
&= x^2 c^{x-1} - \sum_{k=x}^{\infty} k^2 c^{k-1}(1-c)\\
&= \sum_{k=x}^{\infty} x^2 c^{k-1}(1-c) - \sum_{k=x}^{\infty} k^2 c^{k-1}(1-c).\\
&= -2x\sum_{k=x}^{\infty}(k-x)c^{k-1}(1-c) - \sum_{k=x}^{\infty} x^2 c^{k-1}(1-c) + 2\sum_{k=x}^{\infty} kxc^{k-1}(1-c) - \sum_{k=x}^{\infty} k^2 c^{x-1}(1-c)\\
&= -2x\sum_{k=x}^{\infty}(k-x)c^{k-1}(1-c) - \sum_{k=x}^{\infty}(k-x)^2 c^{k-1}(1-c)\\
&= -2x\sum_{k=1}^{\infty}(k-1)c^{k-1+x-1}(1-c) - \sum_{k=1}^{\infty}(k-1)^2 c^{k-1+x-1}(1-c)\\
&= c^{x-1}\Big(-2x\sum_{k=1}^{\infty}(k-1)c^{k-1}(1-c) - \sum_{k=1}^{\infty}(k-1)^2 c^{k-1}(1-c)\Big)\\
&= c^{x-1}(-2x\mathbb{E}[X-1] - \mathbb{E}[(X-1)^2])\\
&= c^{x-1}\frac{-2xc(1-c) - (c+c^2)}{(1-c)^2}
\end{aligned}$$

Next, we look at the term $\mathbb{E}[X]^2 - \mathbb{E}[Var(\mathcal{W}_{i.}|N_i = x)]^2$ in more detail.

$$\mathbb{E}[X]^2 - \mathbb{E}[Var(\mathcal{W}_{i.}|N_i = x)]^2 = \big(\mathbb{E}[X] - \mathbb{E}[Var(\mathcal{W}_{i.}|N_i = x)]\big)\big(\mathbb{E}[X] + \mathbb{E}[Var(\mathcal{W}_{i.}|N_i = x)]\big)$$

$$
\begin{aligned}
\mathbb{E}[X] - \mathbb{E}[Var(\mathcal{W}_{i\cdot}|N_i = x)] &= \sum_{k=1}^{\infty} kc^{k-1}(1-c) - \sum_{k=1}^{x-1} kc^{k-1}(1-c) - x\mathbb{P}(X \geq x) \\
&= \sum_{k=x}^{\infty} kc^{k-1}(1-c) - xc^{x-1} \\
&= \sum_{k=0}^{\infty} (k+x)c^{k-1+x}(1-c) - xc^{x-1} \\
&= \sum_{k=0}^{\infty} kc^{k-1+x}(1-c) + \sum_{k=0}^{\infty} xc^{k-1+x}(1-c) - xc^{x-1} \\
&= c^{x-1}\frac{c}{1-c} + c^{x-1}x - xc^{x-1} \\
&= \frac{c^x}{1-c}
\end{aligned}
$$

Now replacing these back at the original equation we have

$$
\begin{aligned}
Var(\mathcal{W}_{i\cdot}|N_i = x) - Var(X) &= c^{x-1}\frac{-2xc(1-c) - (c+c^2)}{(1-c)^2} + \frac{c^x}{1-c}\Big(2\mathbb{E}[X] - \frac{c^x}{1-c}\Big) \\
&= c^{x-1}\frac{-2xc(1-c) - (c+c^2) + 2c - c(c^{x-1})}{(1-c)^2} \\
&= c^x\frac{-2x(1-c) - 1 - c + 2 - c^{x-1}}{(1-c)^2} \\
&= c^x\frac{(1-c)(-2x+1) - c^{x-1}}{(1-c)^2} < 0.
\end{aligned}
$$

Thus $\mathbb{E}[Var(\mathcal{W}_{i\cdot}|N_i)] - Var(X) < 0$. Furthermore we derive

$$
\mathbb{E}[\mathcal{W}_{i\cdot}|N_i = x] = \sum_{k=1}^{x-1} k\mathbb{P}(X = k) + x\mathbb{P}(X \geq k) = \sum_{k=1}^{x-1} c^{k-1} = \frac{1-c^{x+1}}{1-c},
$$

and thus the variance of $\mathbb{E}[\mathcal{W}_{i\cdot}|N_i = x]$ satisfies

$$
Var(\mathbb{E}[\mathcal{W}_{i\cdot}|N_i = x]) = \frac{c^2 Var(c^x)}{(1-c)^2} \leq \frac{c^4}{(1-c)^2},
$$

since for non-dangling nodes the random variable c^x takes values only in the interval $[0, c]$. So, we have

$$
Var(\mathcal{W}_{i\cdot}) < \frac{c}{(1-c)^2} + \frac{c^2}{(1-c)^2} = \frac{c(1+c^3)}{(1-c)^2},
$$

which completes the proof of our lemma. □

Theorem 2. *Given the event that the estimator $\bar{\mathcal{W}}_{\cdot j}$ satisfies $|\bar{\mathcal{W}}_{\cdot j} - w_{\cdot j}| \leq \varepsilon w_{\cdot j}$, the event $|\bar{\pi}_j - \pi_j| \leq \varepsilon_{n,\beta}\pi_j$ occurs with probability at least $1-\beta$ for any $\beta > 0$ and $\varepsilon_{n,\beta}$ satisfying $|\varepsilon - \varepsilon_{n,\beta}| < \frac{C(\beta)(1+\varepsilon)}{\sqrt{nm}}$. The factor $C(\beta)$ can be approximated as $C(\beta) \approx x_{1-\beta/2}\sqrt{\frac{n-n_0}{n}(1+c^3)}\frac{c}{1-c}$, where $x_{1-\beta/2}$ is a $1-\beta/2$-quantile of the standard normal distribution and n_0 is the number of dangling nodes.*

Proof. Using (12) and (13) we derive

$$\begin{aligned}
\bar{\pi}_j - \pi_j &= \bar{\mathcal{W}}_{.j}\bar{\mathcal{W}}^{-1} - \pi_j \\
&= \bar{\mathcal{W}}_{.j}\bar{\mathcal{W}}^{-1} - w_{.j}\bar{\mathcal{W}}^{-1} + w_{.j}\bar{\mathcal{W}}^{-1} - \pi_j \\
&= \bar{\mathcal{W}}_{.j}\bar{\mathcal{W}}^{-1} - w_{.j}\bar{\mathcal{W}}^{-1} + \gamma^{-1}\bar{\mathcal{W}}^{-1}\pi_j - \pi_j \\
&= \gamma(\gamma\bar{\mathcal{W}})^{-1}(\bar{\mathcal{W}}_{.j} - w_{.j}) + \pi_j((\gamma\bar{\mathcal{W}})^{-1} - 1) \\
&\le \gamma(\gamma\bar{\mathcal{W}})^{-1}\varepsilon w_{.j} + \pi_j((\gamma\bar{\mathcal{W}})^{-1} - 1) \\
&= \varepsilon\pi_j - \varepsilon\pi_j + \gamma(\gamma\bar{\mathcal{W}})^{-1}\varepsilon w_{.j} + \pi_j((\gamma\bar{\mathcal{W}})^{-1} - 1) \\
&= \varepsilon\pi_j + \varepsilon\pi_j((\gamma\bar{\mathcal{W}})^{-1} - 1) + \pi_j((\gamma\bar{\mathcal{W}})^{-1} - 1) \\
&= \varepsilon\pi_j + ((\gamma\bar{\mathcal{W}})^{-1} - 1)(\varepsilon + 1)\pi_j.
\end{aligned}$$

Let us now investigate the term $(\gamma\bar{\mathcal{W}})^{-1}$. First, note that the random variables $\bar{\mathcal{W}}_{i.} = \sum_{j=1}^{n} \bar{\mathcal{W}}_{ij}$, $i \in \mathcal{I}_1$ are independent because they are determined by simulation runs initiated at different pages. Further, for a non-dangling node i, using Lemma 2 we find $\mathbb{E}[\bar{\mathcal{W}}_{i.}] = \sum_{j=1}^{n} w_{ij}$ and $Var(\bar{\mathcal{W}}_{i.}) = \frac{1}{m}Var(\mathcal{W}_{i.}) \le \frac{1}{m}\frac{c(1+c^3)}{(1-c)^2}$.

Thus, $\bar{\mathcal{W}}$ equals the number of non-dangling nodes n_0 plus the sum of $n - n_0$ independent random variables $\bar{\mathcal{W}}_{i.}$, $i \in \mathcal{I}_1$. Since the number $n - n_0$ is obviously very large, $\bar{\mathcal{W}}$ is approximately normally distributed with mean γ^{-1} and variance $Var(\bar{\mathcal{W}}) = \sum_{i\in\mathcal{I}_1} Var(\bar{\mathcal{W}}_{i.}) \le (n - n_0)\frac{c(1+c^3)}{m(1-c)^2}$. Hence, $\gamma\bar{\mathcal{W}}$ is approximately normally distributed with mean 1 and variance

$$Var(\gamma\bar{\mathcal{W}}) \le \gamma^2(n - n_0)\frac{c(1+c^3)}{m(1-c)^2} < \frac{n - n_0}{n^2}\frac{c(1+c^3)}{m(1-c)^2}.$$

Now we look for the $\varepsilon_{n,\beta}$ which will satisfy the statement of our theorem.

$$\begin{aligned}
\mathbb{P}(|\bar{\pi}_j - \pi_j| \le \varepsilon_{n,\beta}\pi_j) &\ge \mathbb{P}(\varepsilon\pi_j + |(\gamma\bar{\mathcal{W}})^{-1} - 1|(1 + \varepsilon)\pi_j \le \varepsilon_{n,\beta}\pi_j) \\
&= \mathbb{P}\big(|(\gamma\bar{\mathcal{W}})^{-1} - 1| \le \frac{|\varepsilon - \varepsilon_{\beta,n}|}{1 + \varepsilon}\big) \\
&= \mathbb{P}\big(|(\gamma\bar{\mathcal{W}})^{-1} - 1| \le \frac{C(\beta)}{\sqrt{nm}}\big).
\end{aligned}$$

Let us consider a $(1-\beta)$-confidence interval defined as $\mathbb{P}(|(\gamma\bar{\mathcal{W}})^{-1} - 1| \le z) \ge 1 - \beta$ for some small positive β and z. If z is small enough so that $1/(1-z) \approx 1 + z$ and $1/(1+z) \approx 1 - z$, then the above probability approximately equals $\mathbb{P}(|\gamma\bar{\mathcal{W}} - 1| \le z)$ and the inequality holds for all z satisfying $z \ge x_{1-\beta/2}\frac{c}{1-c}\sqrt{\frac{n-n_0}{n}(1+c^3)\frac{c}{1-c}}\frac{1}{\sqrt{nm}}$ so that $\mathbb{P}(|(\gamma\bar{\mathcal{W}})^{-1} - 1| \le \frac{C(\beta)}{\sqrt{nm}}) \ge 1 - \beta$ is true for

$$C(\beta) \approx x_{1-\beta/2}\sqrt{\frac{n - n_0}{n}(1 + c^3)\frac{c}{1 - c}}.$$

□

Consider the confidence interval for $\bar{\mathcal{W}}_{.j}$ defined as $\mathbb{P}(|\bar{\mathcal{W}}_{.j} - w_{.j}| < \varepsilon w_{.j}) \ge 1 - \alpha$. From (16) and (17), we have $\mathbb{E}[\bar{\mathcal{W}}_{.j}] = w_{.j}$ and $Var(\bar{\mathcal{W}}_{.j}) \le \frac{1}{m}\frac{1+q_{jj}}{1-q_{jj}}w_{.j}$. Since $\bar{\mathcal{W}}_{.j}$ is a sum of large number of terms, the random variable $(\bar{\mathcal{W}}_{.j} - w_{.j})/\sqrt{Var(\bar{\mathcal{W}}_{.j})}$ has approximately a standard normal distribution. Thus, we have

$$\frac{\varepsilon w_{.j}}{\sqrt{Var(\bar{\mathcal{W}}_{.j})}} \ge x_{1-\alpha/2},$$

which results in

$$m \geq \frac{1+q_{jj}}{1-q_{jj}} \frac{x^2_{1-\alpha/2}}{\varepsilon^2 w_{\cdot j}}. \tag{19}$$

So, as m goes to infinity, condition of the theorem is satisfied and therefore the estimator $\bar{\pi}_j$ converges to π_j in probability. Thus the estimator $\bar{\pi}_j$ is consistent.

The theorem also states that the error in the estimate of π_j originates mainly from estimating $w_{\cdot j}$. The additional relative error caused by estimating γ as $\left(\sum \bar{\mathcal{W}}_{\cdot j}\right)^{-1}$, is of the order $1/\sqrt{mn}$ with arbitrarily high probability, and thus this error can be neglected.

Now, applying $w_{\cdot j} = \gamma^{-1}\pi_j$ to (19), we get

$$m \approx \frac{1-q_{jj}}{1-q_{jj}} \frac{\gamma x_{1-\alpha/2}}{\varepsilon^2 \pi_j}.$$

Note that $\pi_j \geq \gamma$ for all $j = 1...n$. Thus, with a high probability, a couple of hundred iterations allows to evaluate the PageRank of all pages with relative error at most 0.1. In practice, however it is essential to evaluate well the PageRank of important pages in a short time. Therefore, let us evaluate the relative error ε for a given value of π_j. Using (9), we derive

$$\varepsilon \approx x_{1-\alpha/2}\sqrt{\frac{1+q_{jj}}{1-q_{jj}}}\,\frac{\sqrt{1-c+c\sum_{i\in\mathcal{I}_0}\pi_i}}{\sqrt{\pi_j}\sqrt{mn}}. \tag{20}$$

From the examples of PageRank values presented in [2], it follows that the PageRank values of popular pages are at least 10^4 times greater than the PageRank values of average pages. Since the PageRank value is bounded from below by $(1-c)/n$, the formula (20) implies that if the important pages have PageRank values 10^4 times larger than the PageRank of the pages with minimal PageRank value, the Monte Carlo method achieves an error of about %1 for the important pages already after the first iteration.

Let us now compare the precision of the end-point version and the complete path version of the Monte Carlo methods. According to Algorithm 1, the end-point version estimates π_j simply as a fraction of $N = mn$ random walks that end at page j. We can construct a confidence interval as follows:

$$\mathbb{P}(\hat{\pi}_{j,N} - \pi_j|) < \varepsilon\pi_j) = 1 - \alpha.$$

Using again the standard normal distribution, we get

$$\varepsilon = x_{1-\alpha/2}\frac{\sqrt{1-\pi_j}}{\sqrt{\pi_j}\sqrt{mn}}. \tag{21}$$

Forgetting for a moment about slight corrections caused by the trade-off between random and cyclic start, we see that the choice between end-point version and the complete path version essentially depends on two factors: the total PageRank of dangling nodes and the probability of a cycle when a random walk started from j returns back to j. If the Web graph has many short cycles, then the information from registering visits to every page is obtained at cost of high variability which leads to a worse precision. Also if the total rank of dangling nodes is high, then the random walk will often reach dangling nodes and stop. This can have negative impact on the complete path algorithm. The two points mentioned above can make the difference between the end-point and the complete-path versions negligible. However the experiments in [11] indicate that the complete path version is more efficient than the end-point versions when the real Web structure is considered. For further information on numerical experiments, one can look at [11].

We remark that if the results of the first iteration are not satisfactory, it is hard to improve them by increasing m. After m iterations, the relative error of the Monte Carlo method will reduce on average only by factor $1/\sqrt{m}$ whereas the error of the power iteration method decreases exponentially with m. However, because of simplicity in implementation, the Monte Carlo algorithms can be still advantageous even if a high precision is required.

Finally, we would like to emphasize that the Monte Carlo algorithms have natural parallel implementation and they allow to perform a continuous update of the PageRank vector. Google prefers to recompute the PageRank vector starting from the uniform distribution rather than the PageRank vector of the previous month as initial approximation and it takes about a week to compute the new PageRank vector. We suggest to run Monte Carlo algorithm continuously while the database is updated with new data. Hence, we would have an up-to-date estimation of the PageRank for relatively important pages with high accuracy. Then once in a while, one can run the power iteration method to have a good PageRank estimation for all pages.

References

[1] J. Kleinberg, *Authoritative sources in a hyperlinked environment*, Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms (Philadelphia, PA, USA), SODA '98, Society for Industrial and Applied Mathematics, 1998, pp. 668–677.

[2] L. Page and S. Brina and R. Motwani and T. Winograd, *The pagerank citation ranking: Bringing order to the web*, Stanford University Technical Report, 1998.

[3] C. Meyer, *Matrix analysis and applied linear algebra*, SIAM, 2000, 527.

[4] R. Lempel and S. Moran, *Salsa: the stochastic approach for link-structure analysis*, ACM Trans. Inf. Syst. **19** (2001), 131–160.

[5] L.A. Breyer, *Markovian page ranking distributions:some theory and simulations*, Tech. report, 2002, available at http://www.lbreyer.com/preprints.html.

[6] S.D. Kamvar and T.H. Haveliwala and C.D. Manning and G.H. Golub, *Adaptive methods for the computation of pagerank*, Tech. report, Stanford University, 2003.

[7] ———, *Extrapolation methods for accelerating pagerank computations*, 2003, pp. 261–270.

[8] A.N. Langville and C. Meyer, *Survey: Deeper inside pagerank*, Internet Mathematics **1** (2003), no. 3, 335–380.

[9] ———, *Updating pagerank with iterative aggregation*, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (New York, NY, USA), ACM, 2004, pp. 392–393.

[10] K. Bryan and T. Leise, *The $25,000,000,000 eigenvector: the linear algebra behind google*, SIAM Review **48** (2006), 569–581.

[11] K. Avrachenkov and N. Litvak and D. Nemirovsky and N. Osipova, *Monte carlo methods in pagerank computation: When one iteration is sufficient*, SIAM J. Numer. Anal. **45** (2007), 890–904.

[12] A. Bonato, *A course on the web graph*, American Mathematical Society, Boston, MA, USA, 2008, 97–110.