Using Polynomial Chaos to Compute the Influence of Multiple Random Surfers in the PageRank Model

Paul G. Constantine and David F. Gleich

Stanford University
Institute for Computational and Mathematical Engineering
{paul.constantine,dgleich}@stanford.edu

Abstract. The PageRank equation computes the importance of pages in a web graph relative to a single random surfer with a constant teleportation coefficient. To be globally relevant, the teleportation coefficient should account for the influence of all users. Therefore, we correct the PageRank formulation by modeling the teleportation coefficient as a random variable distributed according to user behavior. With this correction, the PageRank values themselves become random. We present two methods to quantify the uncertainty in the random PageRank: a Monte Carlo sampling algorithm and an algorithm based the truncated polynomial chaos expansion of the random quantities. With each of these methods, we compute the expectation and standard deviation of the PageRanks. Our statistical analysis shows that the standard deviation of the PageRanks are uncorrelated with the PageRank vector.

1 Introduction

In its purest form, the PageRank model ignores the text underlying pages on the web and creates an irreducible, aperiod Markov chain model for a hypothetical random surfer on the link structure of the web [1]. Each entry of the stationary distribution measures the global importance of a page.

The PageRank model, however, is not unique. A PageRank value depends upon a parameter α which controls how the putative random surfer "teleports" around the web. Upon visiting a website, the random surfer chooses an outlink uniformly at random with probability α and chooses a page according to a prior distribution with probability $1 - \alpha$. This paper focuses on the modeling assumptions for the value of α and suggests a new model for PageRank that fixes a modeling error in the original PageRank formulation.

To continue our discussion, we must define the PageRank model and establish some notation. Let $\mathbf{W}$ be an adjacency matrix for a web graph, $w_{i,j} = 1$ when node i links to node j. We set $\mathbf{P}$ to be a fully row-stochastic random walk transition matrix on $\mathbf{W}$. The matrix $\mathbf{P}$ has dangling nodes corrected in an arbitrary way (for example, see [2,3]) such that $\mathbf{Pe} = \mathbf{e}$ where $\mathbf{e}$ is the vector of all ones. Let $1 - \alpha$ be the teleportation probability and $\mathbf{v}$ be the personalization

A. Bonato and F.R.K. Chung (Eds.): WAW 2007, LNCS 4863, pp. 82–95, 2007.

distribution. The PageRank model requires that $0 \leq \alpha < 1$, $v_i \geq 0$, and $\mathbf{v}^T\mathbf{e} = 1$. With these definitions, the PageRank vector $\mathbf{x}(\alpha)$ is the unique eigenvector with $||\mathbf{x}(\alpha)||_1 = 1$ satisfying

$$\left[\alpha\mathbf{P}^T + (1-\alpha)\mathbf{v}\mathbf{e}^T\right]\mathbf{x}(\alpha) = \mathbf{x}(\alpha) \tag{1}$$

or equivalently [4,5] the solution of the linear system

$$\left(\mathbf{I} - \alpha\mathbf{P}^T\right)\mathbf{x}(\alpha) = (1-\alpha)\mathbf{v}. \tag{2}$$

The key error in the PageRank model is that it only accounts for a single surfer because it only permits a single value of α. The choice of α is quite mysterious. Most researchers take $\alpha = 0.85$ [6]. Recently, Avrachenkov et al. suggested choosing $\alpha = 1/2$ [7]. Their suggestion follows from graph theoretic properties of the PageRank solution vector as a function of α. If we believe the PageRank random surfer model, then α should be estimated from Internet usage logs, so that $\alpha = \mathrm{E}[A]$ where A is a random variable representing the teleportation parameter for each user. We are not aware of any studies that attempt to determine α using this methodology.

However, assuming $\alpha = \mathrm{E}[A]$ does not yield the "correct" PageRank vector This fact follows because in general $\mathrm{E}[\mathbf{x}(A)] \neq \mathbf{x}(\mathrm{E}[A])$. Intuitively, this issue arises because the PageRank model consolidates everyone into a single user. Appendix A demonstrates a formal counterexample. A more realistic model would consider that each user should have a small contribution to the final PageRank values.

To reiterate, computing $\mathbf{x}(\mathrm{E}[A])$ does not yield a PageRank vector that expresses all of the users. Instead, we propose using $\mathrm{E}[\mathbf{x}(A)]$ as a new PageRank vector that accurately models the underlying user population.

While our model for PageRank using a random parameter better represents the reality of random surfers, we would not expect the rankings generated by the model to be qualitatively different from those generated by the approximation of using $\alpha = \mathrm{E}[A]$. We expect $\mathrm{E}[\mathbf{x}(A)] \approx \mathbf{x}(\mathrm{E}[A])$ for "reasonable" distributions of A. Our results confirm this expectation, which justifies use of the PageRank vector as a global ranking for all users.

However, by modeling each component of the PageRank vector as a random variable, we gain a distinct advantage when quantifying the importance of a page. Namely, we can compute the *standard deviation* of each PageRank value with respect to the distribution of A. The standard deviation is a key tool in uncertainty quantification and allows us to examine the pages *most sensitive* to changes in PageRank based on the underlying distribution of A. In the results section, we employ the standard deviation of the PageRank vector to generate rankings that are uncorrelated with the original PageRank vector. Uncorrelated vectors are important because they provide additional useful input to a machine learning framework for generating a web search ranking function.

2 Choice of Distribution

One of the mysteries in the original PageRank model was the choice of α. In our new model, we replace α with a random variable A. Immediately, we face a new question: what should the distribution of A be? This question is much easier to answer! We estimate a value $\hat{\alpha}_i$ from usage statistics for each user i and compute the resulting discrete distribution. Unfortunately, this represents a daunting computational and statistical analysis task.

We do not attempt to determine the underlying discrete distribution and proceed to the limiting case where A is a continuous random variable with support in the interval $[0, 1]$. There are two distributions that potentially model the user behavior of interest: the uniform distribution over $[l, r]$ with $0 \leq l < r \leq 1$ and the Beta distribution with parameters a and b, $a, b \geq -1$. The density functions are

$$f_{U[l,r]}(x) = \frac{1}{r-l} I_{[l,r]}(x) \text{ and } f_{\text{Beta}(a,b)}(x) = \frac{x^b(1-x)^a}{\mathrm{B}(a+1, b+1)} I_{[0,1]}(x),$$

where $\mathrm{B}(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$ and $\Gamma(x)$ is the Euler Γ function. The Beta density becomes the uniform density with $l = 0$ and $r = 1$ when $a = b = 0$. (This form of the Beta density may differ from other presentations. In particular, the Beta distribution implemented in Matlab assumes the uniform distribution when $a = b = 1$.)

3 A Consequence of the Modeling Change

Recall that our proposed change in the PageRank model is to replace the deterministic parameter α with a random variable A and to use $\mathrm{E}[\mathbf{x}(A)]$ instead of $\mathbf{x}(\mathrm{E}[A])$ as the PageRank vector. One attractive feature of this change is that using $\mathrm{E}[\mathbf{x}(A)]$ incorporates more influence from longer paths in the graph. Appendix A derives (5),

$$\mathrm{E}[\mathbf{x}(A)] = \sum_{n=0}^{\infty} \mathrm{E}[A^n - A^{n+1}] \mathbf{P}^{T^n} \mathbf{v}.$$

The coefficient $\mathrm{E}[A^n - A^{n+1}]$ expresses the *weight* placed on paths of length n in the graph. Following Baeza-Yates et al., we call these coefficients the *path damping coefficients* [8]. Figure 1 shows these coefficients as functions of n along with the path damping coefficients in the deterministic case. Appendix A, then, demonstrates that the TotalRank model proposed in that paper is equivalent to using $\mathrm{E}[\mathbf{x}(A)]$ in our model with A distributed uniformly over $[0, 1]$.

4 Computing the Solution

Given the randomness in the PageRank model introduced by the random parameter A, our objective is to quantify the uncertainty in the solution $\mathbf{x}(A)$

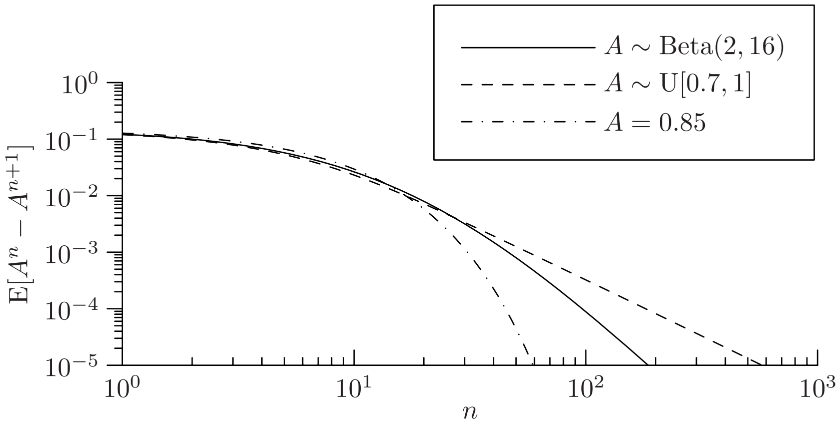


Fig. 1. Different choices of the underlying distribution of A have significant consequences for the influence of long paths in the final PageRank vector. From formula (5) the influence of a path of length n is $\mathrm{E}[A^n - A^{n+1}]$ which we call the path damping coefficient. Assuming that A is uniform puts the most weight on long paths. The three distributions displayed in this plot satisfy $\mathrm{E}[A] = 0.85$. All the methods put similar weight on paths up to length 10.

of the system. In concrete terms, this amounts to computing the mean of the stationary distribution $\mathrm{E}[\mathbf{x}(A)]$ as well as its standard deviation $\mathrm{Std}[\mathbf{x}(A)]$.

4.1 Monte Carlo Approach

One straightforward way to compute these quantities is to use a Monte Carlo approach. First generate M realizations of A from a chosen distribution, and then solve each resulting PageRank problem. With the M different realizations of $\mathbf{x}_i(A)$, $i = 1, \ldots, M$, we can compute unbiased estimates for $\mathrm{E}[\mathbf{x}(A)]$ and $\mathrm{Var}[\mathbf{x}(A)]$ with the formulas

$$\mathrm{E}[\mathbf{x}(A)] \approx \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \equiv \hat{\mu}_{\mathbf{x}}, \quad \mathrm{Std}[\mathbf{x}(A)] \approx \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{x}_i - \hat{\mu}_{\mathbf{x}})^2}$$

from [9]. Unfortunately, as with any Monte Carlo method, these estimates converge as $1/\sqrt{M}$ [10], which makes this approach prohibitively expensive for large systems such as the web graph.

4.2 The Polynomial Chaos Approach

A more efficient way to compute $\mathrm{E}[\mathbf{x}(A)]$ and $\mathrm{Std}[\mathbf{x}(A)]$ — and the one that we advocate in this paper — employs a technique known as the stochastic Galerkin method. This technique utilizes a specific representation of the random parameter A and random response vector $\mathbf{x}(A)$ called the polynomial chaos expansion

(PCE). The PCE expresses the random quantities as an infinite series of orthogonal polynomials that take a vector of random variables as arguments. This representation has its roots in the work of Wiener [11] who expressed a Gaussian process as an infinite series of Hermite polynomials. In the early 1990s, Ghanem and Spanos [12] truncated Wiener's representation to finitely many terms and used this truncated PCE as a primary component of their stochastic finite element method; this truncation made computations possible. In 2002, Xiu and Karniadakis [13] expanded this method to non-Gaussian processes via the more general Wiener-Askey scheme of orthogonal polynomials. The use of the stochastic Galerkin method has become fashionable in the uncertainty quantification community as a technique for measuring the effect of random inputs on partial differential equation models [14,15,16]. This paper presents a straightforward application of this technique to a linear system with a single random parameter.

To introduce the method, let $\{\Psi_k(\boldsymbol{\xi}(\omega))\}, k \in \mathbb{N}$ denote a set of orthogonal polynomials where $\boldsymbol{\xi}(\omega)$ is a vector of i.i.d. random variables. We use the dependence on ω to represent a random quantity. Assume $\{\Psi_k(\boldsymbol{\xi}(\omega))\}$ have the following properties:

$$\mathrm{E}[\Psi_0] = 1, \quad \mathrm{E}[\Psi_k] = 0 \text{ for } k > 0, \quad \mathrm{E}[\Psi_j \Psi_k] = \delta_{jk} \text{ for } j, k \geq 0$$

where δ_{jk} is the Kronecker delta. The PCE of a random quantity $u(\omega)$ is given by

$$u(\omega) = \sum_{k=0}^{\infty} u_k \Psi_k(\boldsymbol{\xi}(\omega))$$

where $\{u_i\}$ are the PCE coefficients. By the Cameron-Martin theorem [17], this series converges in an L_2 sense, i.e.

$$\mathrm{E}\left[\left(u - \sum_{k=0}^{N} u_k \Psi_k(\boldsymbol{\xi}(\omega))\right)^2\right] \to 0$$

as $N \to \infty$. The L_2 convergence of this expansion motivates truncating the series at a finite number of terms for the sake of computation. Thus we can approximate u with the finite series

$$u(\omega) \approx \sum_{k=0}^{N} u_k \Psi_k(\boldsymbol{\xi}(\omega)).$$

Then the problem of computing $u(\omega)$ transforms into the problem of finding the coefficients of its truncated PCE. From this point onward, we drop the explicit dependence on ω in our notation.

Since our particular model has only one random variable, we can fully account for this single random dimension by letting $\boldsymbol{\xi}$ have only one component, which we denote by $\boldsymbol{\xi} = \xi$. For the random parameter A and response quantity $\mathbf{x}$ in our model, we can write their respective PCEs as

$$A = \sum_{k=0}^{\infty} A_k \Psi_k(\xi), \qquad \mathbf{x} = \sum_{k=0}^{\infty} \mathbf{x}_k \Psi_k(\xi),$$

For $A \sim \mathrm{Beta}(a,b)$, we can achieve exponential convergence in the coefficients $\mathbf{x}_k$ by choosing $\{\Psi_k\}$ to be the 1-D Jacobi polynomials with parameters a and b [13]. Note that when $a = b = 0$, $A \sim U[0,1]$ and the Jacobi polynomials reduce to the Legendre polynomials.

Now we substitute these representations into our model,

$$\left(\mathbf{I} - \sum_{k=0}^{\infty} A_k \Psi_k(\xi) \mathbf{P}^T\right) \left(\sum_{k=0}^{\infty} \mathbf{x}_k \Psi_k(\xi)\right) = \left(1 - \sum_{k=0}^{\infty} A_k \Psi_k(\xi)\right) \mathbf{v}.$$

To make this problem amenable to computation, we truncate the PCEs to N terms,

$$\left(\mathbf{I} - \sum_{k=0}^{N} A_k \Psi_k(\xi) \mathbf{P}^T\right) \left(\sum_{k=0}^{N} \mathbf{x}_k \Psi_k(\xi)\right) = \left(1 - \sum_{k=0}^{N} A_k \Psi_k(\xi)\right) \mathbf{v}.$$

In the next section we perform a convergence study on the order of the expansion for our particular application, Fig. 2.

With the distribution of A known explicitly, we can solve directly for A_k. By multiplying both sides of the truncated PCE representation of A by Ψ_j and taking the expectation, the orthogonality of $\{\Psi_k\}$ gives the formula

$$A_j = \frac{\mathrm{E}[A\Psi_j(\xi)]}{\mathrm{E}[\Psi_j(\xi)^2]}, \quad j = 0, \ldots, N.$$

From this formula, we have that $A_0 = \mathrm{E}[A]$ and $A_j = 0$ for $j \geq 2$. Thus our system reduces to

$$(\mathbf{I} - (A_0 + A_1\Psi_1)\mathbf{P}^T) \left(\sum_{k=0}^{N} \mathbf{x}_k \Psi_k(\xi)\right) = (1 - (A_0 + A_1\Psi_1))\mathbf{v}.$$

Multiplying both sides by Ψ_j and taking the expectation, the orthogonality of $\{\Psi_k\}$ leaves

$$\begin{aligned} \mathrm{E}[\Psi_j^2]\mathbf{I} - \sum_{k=0}^{N} \mathrm{E}[(A_0 + A_1\Psi_1)\Psi_j\Psi_k]\mathbf{P}^T\mathbf{x}_k \\ = ((1 - A_0)\,\mathrm{E}[\Psi_j] - A_1\,\mathrm{E}[\Psi_j\Psi_1])\mathbf{v} \end{aligned} \tag{3}$$

for $j = 0 \ldots N$. Therefore we have $N+1$ coupled linear systems to solve for $\mathbf{x}_j$. Note that the dimension of this larger system is $N+1$ times the dimension of $\mathbf{P}$.

Once we solve for the PCE coefficients $\mathbf{x}_j$, we can compute the mean of the PageRank vector,

$$\mathrm{E}[\mathbf{x}] \;=\; \mathrm{E}\left[\sum_{j=0}^{N} \mathbf{x}_j \Psi_j\right] \;=\; \mathbf{x}_0 \underbrace{\mathrm{E}[\Psi_0]}_{=1} + \sum_{j=0}^{N} \mathbf{x}_j \underbrace{\mathrm{E}[\Psi_i]}_{=0} \;=\; \mathbf{x}_0.$$

To compute the standard deviation, we first compute the variance.

$$\begin{aligned}\mathrm{Var}[\mathbf{x}] &= \mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}])^2] \\ &= \mathrm{E}\left[\left(\left(\sum_{j=0}^{N} \mathbf{x}_j \Psi_j\right) - \mathbf{x}_0\right)^2\right] \\ &= \sum_{j=1}^{N} \mathbf{x}_j^2\, \mathrm{E}[\Psi_j^2]; \qquad \text{(by orthogonality)}\end{aligned}$$

and then $\mathrm{Std}[\mathbf{x}] = \sqrt{\mathrm{Var}[\mathbf{x}]}$ is computed element-wise.

5 Datasets

Our experimental datasets came from four sources [18,19,20,21]. We downloaded and modified two datasets compressed using the Webgraph framework [22]. Table 1 summarizes our datasets.

From the webbase dataset, we extracted the web graph corresponding to the `http://cs.stanford.edu` host and computed the largest strongly connected component of this graph. We also computed and used the largest strongly connected component of the cnr-2000 graph. The wikipedia graph is discussed in Sect. 5.1, and the us2004 comes from [20]. Each of the graphs stanford-cs, cnr-2000, and wikipedia is a largest strongly connected component and has a natural random walk

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}, \tag{4}$$

Here $\mathbf{D}$ is the diagonal matrix of outdegrees for each node, and $\mathbf{Pe} = \mathbf{e}$. The us2004 graph was not strongly connected and we added self loops to all dangling nodes before computing $\mathbf{P}$ according to (4). The graphs for stanford-cs and wikipedia, prior to extracting the largest connected component, are available in the University of Florida Sparse Matrix Collection as `Gleich/wb-cs-stanford` and `Gleich/wikipedia-20051105` [23].

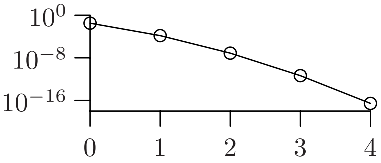
5.1 Wikipedia

On a semi-regular basis, Wikipedia provides a dump of their database. We collected the dump from November 5, 2005 [21] and processed the results into a graph by identifying all links between Wikipedia articles in the text. We decided to remove many pages that were not articles because we wanted the results to be true to the underlying Encyclopedic nature of Wikipedia and felt that pages in the “User” and “User talk” categories did not meet that requirement. The categories we kept were “Category” and “Portal” because they represent overviews

Table 1. The datasets used in this paper vary in scale over three orders of magnitude. The term $id(v)$ is used to represent the indegree of a vertex. Each of the first three graphs listed is a strongly connected component from a larger graph. The final graph is not strongly connected and has dangling nodes adjusted by adding a self-loop.

Name	$\|\mathbf{V}\|$	$\|\mathbf{E}\|$	$\max \mathbf{id}(\mathbf{v})$	**Source**
stanford-cs	2,759	13,895	340	[18,22]
cnr-2000	112,023	1,646,332	18,235	[19,22]
wikipedia	1,103,453	18,245,140	71,524	Sect. 5.1
us2004	6,411,252	23,940,956	116,393	[20]

N	**stanford-cs**	**cnr-2000**
0	3.29×10^{-2}	3.20×10^{-2}
1	1.48×10^{-4}	1.47×10^{-4}
2	7.56×10^{-8}	7.54×10^{-8}
3	4.46×10^{-12}	4.47×10^{-12}
4	2.31×10^{-17}	2.75×10^{-17}



Values of $\left\|\mathrm{E}[\mathbf{x}_{(N+1)}(A)] - \mathrm{E}[\mathbf{x}_{(N)}(A)]\right\|_1$

Fig. 2. The order study for the polynomial chaos expansion shows that a fourth order expansion is sufficient and that the convergence of the expansion is independent of the graph size; the plots are indistinguishable and we have plotted the data from cnr-2000

of different areas of the Encyclopedia. Finally, we removed all pages in the graph not in the largest strongly connected component. The result of this processing is our wikipedia dataset.

6 Convergence Results

We conducted simple convergence studies with $A \sim \text{Beta}(2, 16)$ on our two small datasets, stanford-cs and cnr-2000, that motivate choices used on the results for our larger datasets. From Figs. 2 and 3, we observed both the predicted

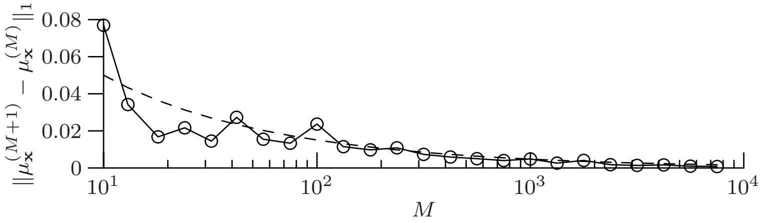


Fig. 3. Monte Carlo sampling theory predicts that the estimates converge proportional to $1/\sqrt{M}$. In this figure, we demonstrate this convergence on our problem. The dashed curve is $0.15/\sqrt{M}$ where the scaling was matched by hand to the underlying data.

exponential convergence of the PCE coefficients and the characteristic slow convergence of the Monte Carlo algorithm. Therefore, we use a fourth order PCE truncation ($N = 4$) in the our subsequent computations and do not consider Monte Carlo as a feasible computational algorithm. Appendix B describes the implementation for these experiments.

7 Results and Discussion

Tables 2-5 present our main results. In the experiments, we computed $\mathrm{E}[\mathbf{x}(A)]$ and $\mathrm{Std}[\mathbf{x}(A)]$ under the following modeling assumptions: (1) $A = 0.85$ is deterministic; (2) A has a Beta distribution with $a = 2$, $b = 16$; and (3) A is distributed uniformly over $[0.7, 1]$. We chose these two distributions because $\mathrm{E}[A] = 0.85$ in both cases. The two graphs used for experiments are wikipedia and us2004. In the remainder of this section, we will adopt the notation that $\hat{\mathbf{x}}_\alpha = \mathbf{x}(0.85)$, $\hat{\mathbf{x}}_{\mathrm{Beta}} = \mathrm{E}[\mathbf{x}(A)]$ where $A \sim \mathrm{Beta}(2, 16)$, and $\hat{\mathbf{x}}_U = \mathrm{E}[\mathbf{x}(A)]$ where $A \sim U[0.7, 1]$. Similarly, $\hat{\mathbf{s}}_{\mathrm{Beta}} = \mathrm{Std}[\mathbf{x}(A)]$ and $\hat{\mathbf{s}}_U = \mathrm{Std}[\mathbf{x}(A)]$ for the respective distributions.

Table 2 presents the time required for the major computational task in each evaluation. For the PageRank problem with a deterministic α, the major computational task is the iterative method used to solve the linear system (2). For the problem with a random variable A, the major computational task is solving the coupled linear systems for the coefficients of the PCE (3). The time required to compute the PCE coefficients is approximately 100 times larger than the time required to compute the PageRank vector. This implies that using our codes, we have an allowance of only 100 Monte Carlo samples before the PCE approach becomes more efficient. The computational codes for the PCE coefficients are not optimized. Therefore, using well known results and techniques from numerical linear algebra (for example [24]) will yield a substantial improvement in these computational times.

Next, we evaluated the top 10 pages for each of the ordering induced by $\hat{\mathbf{x}}_\alpha$, $\hat{\mathbf{x}}_{\mathrm{Beta}}$, and $\hat{\mathbf{x}}_U$. Unsurprisingly, these groups of pages are identical. We evaluate the difference between these three vectors in Tab. 3 using the 1-norm, ∞-norm, and Kendall-τ correlation coefficient. The results in the table clearly demonstrate that while $\mathrm{E}[\mathbf{x}(A)] \neq \mathbf{x}(\mathrm{E}[A])$, $\mathrm{E}[\mathbf{x}(A)] \approx \mathbf{x}(\mathrm{E}[A])$. From the strong τ correlation coefficients, the rankings induced by these vectors are nearly indistinguishable. These results justify using a deterministic approximation of PageRank to the underlying model where A is a random variable.

The next set of our results concerns the standard deviation of the PageRank vector with respect to the distribution of A. Table 4 displays the top 10 pages in each graph according to the PageRank vector and the top 10 pages with highest standard deviation under both distributions. The results first show that countries and years make up the most important pages in Wikipedia according to the PageRank model. Next, the pages with highest standard deviation are the category pages if A is sampled from a Beta distribution. This

Table 2. The time required to solve the major computational task in each model

	wikipedia	us2004
$A = \alpha = 0.85$	60.4 sec.	25.2 sec.
$A \sim \text{Beta}(2, 16)$	2210 sec.	3248 sec.
$A \sim U[0.7, 1.0]$	1936 sec.	2712 sec.

Table 3. In this table, $\hat{\mathbf{x}}_\alpha$ represents $\mathbf{x}(0.85)$, $\hat{\mathbf{x}}_U$ represents $\text{E}[\mathbf{x}(A)]$, $A \sim U[0.7, 1]$, and $\hat{\mathbf{x}}_{\text{Beta}} = \text{E}[\mathbf{x}(A)], A \sim \text{Beta}(2, 16)$. We present the difference between each of these vectors in the 1-norm, ∞-norm, and Kendall-τ correlation coefficient. These results show that the PageRank vector with a deterministic α is extremely close to the expected PageRank vectors for random variable A. In particular, the τ results indicate the rankings induced by each of the vectors are almost identical.

	wikipedia			us2004		
$\mathbf{y}, \mathbf{z}$	$\lvert\lvert\mathbf{y} - \mathbf{z}\rvert\rvert_1$	$\lvert\lvert\mathbf{y} - \mathbf{z}\rvert\rvert_\infty$	$\tau(\mathbf{y}, \mathbf{z})$	$\lvert\lvert\mathbf{y} - \mathbf{z}\rvert\rvert_1$	$\lvert\lvert\mathbf{y} - \mathbf{z}\rvert\rvert_\infty$	$\tau(\mathbf{y}, \mathbf{z})$
$\hat{\mathbf{x}}_\alpha, \hat{\mathbf{x}}_U$	0.04996	0.00018	0.99997	0.04996	0.00013	0.99999
$\hat{\mathbf{x}}_\alpha, \hat{\mathbf{x}}_{\text{Beta}}$	0.03211	0.00012	0.99702	0.03209	8×10^{-5}	0.99872
$\hat{\mathbf{x}}_U, \hat{\mathbf{x}}_{\text{Beta}}$	0.01792	5×10^{-5}	0.99705	0.01807	5×10^{-5}	0.99873

implies that these pages have the highest uncertainty in their ranking. A machine learning framework could theoretically use this additional information with user reviews of pages to generate more accurate rankings. Alternatively, pages with high standard deviation might make good suggestions if a search algorithm got a second chance to present results, given that the first set were unsatisfactory. In the second case, category pages are a logical set of suggestions. We were unable to determine any pattern to the pages with highest standard deviation if A is sampled from a uniform distribution. For the us2004 graph, the pages with highest PageRank tend to have high standard deviation as well.

We now attempt to analyze the standard deviation vectors using the Kendall-τ ranking correlation. Table 5 presents the correlation coefficients. The major result of this table is that the standard deviation vector for both distributions on the wikipedia graph is uncorrelated with any of the PageRank vectors whereas the standard deviation for both distributions on the us2004 graph are anti-correlated. When we evaluate the rank correlation in the us2004 graph with dangling nodes removed, then the rank correlation is positive. We believe that the difference between these results is a consequence of the structural differences between the graphs. The wikipedia graph is the largest strongly connected component of a larger graph and there are no dangling nodes. We believe that the dangling nodes in the us2004 graph are the cause of the strong anti-correlation between the standard deviation and the PageRank vector. The considerable change in the rank-correlation after removing the dangling nodes supports this hypothesis.

Table 4. This table compares the 10 best pages from each of the two experimental graphs with the 10 pages of highest standard deviation under each distributions for the random variable A. While each set of pages is significantly different in the wikipedia graph, the corresponding sets for the us2004 graph are similar.

wikipedia

Top 10 by PageRank	**Largest std for Beta**$(2,16)$	**Largest std for** $U[0.7,1.0]$
United States	Category:Wikiportals	2000
Race (U.S. Census)	Category:Politics	Square kilometer
United Kingdom	Category:Categories	Population density
France	Category:Culture	Race (U.S. Census)
2005	Category:Geography	Per capita income
2004	Category:Countries	Poverty line
2000	Category:Human societies	Census
Canada	Race (U.S. Census)	Marriage
England	Category:Categories by country	Square mile
Category:Categories by country	Category:North American countries	Category:Categories by country

us2004

Top 10 by PageRank	**Largest std for Beta**$(2,16)$	**Largest std for** $U[0.7,1.0]$
lxr.linux.no/	lxr.linux.no/	lxr.linux.no/
kernel.org/	kernel.org/	kernel.org/
examples.oreilly.com/linuxdrive2/	examples.oreilly.com/linuxdrive2/	examples.oreilly.com/linuxdrive2/
www.fsmlabs.com/[...]/openrtlinux/	www.fsmlabs.com/[...]/openrtlinux/	www.fsmlabs.com/[...]/openrtlinux/
www.datadosen.se/jalbum	www.datadosen.se/jalbum	www.datadosen.se/jalbum
linguistics.buffalo.edu/ssila/index.htm	www.iub.edu/	www.indiana.edu/copyright.html
www.iub.edu/	www.indiana.edu/&	www.indiana.edu/
www.uiowa.edu/ ournews/index.html	www.indiana.edu/copyright.html	registrar.indiana.edu/[...]/index.html
catalog.arizona.edu/	www.indiana.edu/	www.uky.edu/
validator.w3.org/check/referer	www.uiowa.edu/õurnews/index.html	www.arizona.edu/

Table 5. This table shows the Kendall-τ ranking correlation coefficient between the standard deviation of the PageRank vector under both distributions for A and the expectations of the PageRank vector. See the text for the interpretation of the vectors in the table. For both the uniformly and Beta distributed random variables, the standard deviation of the wikipedia vector is uncorrelated with the PageRank vector itself. In contrast, the standard deviation of the us2004 vectors are anti-correlated with the PageRank vectors. The label us2004nd indicates the vector from the us2004 graph restricted to the non-dangling pages. On the non-dangling pages, we observe a more mild positive correlation between the PageRank vector and the standard deviation.

	wikipedia	us2004	us2004nd
$\tau(\hat{\mathbf{s}}_U, \hat{\mathbf{x}}_\alpha)$	-0.0032	-0.7846	0.4611
$\tau(\hat{\mathbf{s}}_U, \hat{\mathbf{x}}_U)$	-0.0032	-0.7846	0.4611
$\tau(\hat{\mathbf{s}}_U, \hat{\mathbf{x}}_{\mathrm{Beta}})$	-0.0020	-0.7851	0.4625
$\tau(\hat{\mathbf{s}}_{\mathrm{Beta}}, \hat{\mathbf{x}}_\alpha)$	0.0488	-0.7909	0.5920
$\tau(\hat{\mathbf{s}}_{\mathrm{Beta}}, \hat{\mathbf{x}}_U)$	0.0488	-0.7909	0.5920
$\tau(\hat{\mathbf{s}}_{\mathrm{Beta}}, \hat{\mathbf{x}}_{\mathrm{Beta}})$	0.0021	-0.7918	0.5933

8 Conclusions and Future Work

The PageRank model for computing a global importance vector over web pages makes a critical modeling error by assuming that all users can be represented by a single teleportation parameter. We propose a new model for PageRank where the teleportation coefficient is a random variable supported on the interval $[0,1]$.

Using a truncated polynomial chaos expansion to represent the random teleportation coefficient and PageRank, we compute the expectation and standard deviation of the PageRank vector for two distributions of the random variable in the PageRank model. Our results indicate two important conclusions. First, the expectation of the PageRank model assuming a random variable for the teleportation coefficient yields almost the same ranking as the PageRank model assuming a deterministic teleportation coefficient. This result justifies using the deterministic approximation as a global ranking vector for all users. Second, the standard deviation of the PageRank vector can be uncorrelated with the PageRank vector itself.

There is a significant amount of remaining work to fully investigate the use of the new model for PageRank. First, we need to investigate other distributions for the teleportation parameter, particularly distributions based on statistical analysis of actual usage data. Also, we suspect that the time required to compute the PCE coefficients can be significantly reduced. Additionally, we need to continue to investigate the impact of dangling nodes and strongly connected components on the standard deviation of the PageRank. We hypothesize that there is a significant relationship between these factors.

References

1. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford University (1999)
2. Jeh, G., Widom, J.: Scaling personalized web search. In: Proceedings of the 12th international conference on the World Wide Web, Budapest, Hungary, pp. 271–279. ACM, New York (2003)
3. Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H.: Extrapolation methods for accelerating PageRank computations. In: Proceedings of the 12th international conference on the World Wide Web, pp. 261–270. ACM Press, New York (2003)
4. Arasu, A., Novak, J., Tomkins, A., Tomlin, J.: PageRank computation and the structure of the web: Experiments and algorithms. In: Proceedings of the 11th international conference on the World Wide Web (2002)
5. Del Corso, G.M., Gullí, A., Romani, F.: Fast PageRank computation via a sparse linear system. Internet Mathematics 2(3), 251–273 (2005)
6. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press (2006)
7. Avrachenkov, K., Litvak, N., Pham, K.S.: A singular perturbation approach for choosing PageRank damping factor. arXiv e-prints (2006)
8. Baeza-Yates, R., Boldi, P., Castillo, C.: Generalizing PageRank: Damping functions for link-based ranking algorithms. In: Proceedings of ACM SIGIR, Seattle, Washington, USA, pp. 308–315. ACM Press, New York (2006)
9. Rice, J.A.: Mathematical Statistics and Data Analysis, 2nd edn. Duxbury Press, Boston (1995)
10. Ripley, B.D.: Stochastic Simulation, 1st edn. Wiley, Chichester (1987)
11. Wiener, N.: The homogeneous chaos. American Journal of Mathematics 60, 897–936 (1938)

12. Ghanem, R.G., Spanos, P.D.: Stochastic Finite Elements: A Spectral Approach, 1st edn. Springer, New York (1991)
13. Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. SIAM J. Sci. Comput. 24(2), 619–644 (2002)
14. Babuška, I., Tempone, R., Zouraris, G.: Galerkin finite element approximations of stochastic elliptic differential equations. SIAM Journal of Numeical Analysis 42(2), 800–825 (2004)
15. Wan, X., Karniadakis, G.E.: An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. Journal of Computational Physics 209, 617–642 (2005)
16. Maître, O.P.L., Knio, O.M., Najm, H.N., Ghanem, R.G.: A stochastic projection method for fluid flow. Journal of Computational Physics 173, 481–511 (2001)
17. Cameron, R.H., Martin, W.T.: The orthogonal development of non-linear functionals in series of fourier-hermite functionals. The Annals of Mathematics 48, 385–392 (1947)
18. Hirai, J., Raghavan, S., Garcia-Molina, H., Paepcke, A.: WebBase: a repository of web pages. Computer Networks 33(1-6), 277–293 (2000)
19. Boldi, P., Codenotti, B., Santini, M., Vigna, S.: UbiCrawler: A scalable fully distributed web crawler. Software: Practice & Experience 34(8), 711–726 (2004)
20. Thelwall, M.: A free database of university web links: Data collection issues. International Journal of Scientometrics, Informetrics and Bibliometrics 6/7(1) (2003)
21. Various: Wikipedia XML database dump from November 5, 2005 (November 2005), Accessed from `http://en.wikipedia.org/wiki/Wikipedia:Database_download`
22. Boldi, P., Vigna, S.: Codes for the world wide web. Internet Mathematics 2, 407–429 (2005)
23. Davis, T.: University of Florida sparse matrix collection. NA Digest, vol. 92(42), October 16, 1994, NA Digest, vol. 96(28), July 23, 1996, and NA Digest, vol. 97(23), June 7, 1997 (2007), `http://www.cise.ufl.edu/research/sparse/matrices/`
24. Golub, G.H., van Loan, C.F.: Matrix Computations (Johns Hopkins Studies in Mathematical Sciences). The Johns Hopkins University Press, Baltimore (1996)

A A Counterexample

We demonstrate $\mathrm{E}[\mathbf{x}(A)] \neq \mathbf{x}(\mathrm{E}[A])$ with a counterexample. Set

$$P = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

and $v = \left[1/3\ 1/3\ 1/3\right]^T$. If A is a uniform random variable on $[0, 1]$, then $\mathrm{E}[A] = 1/2$ and

$$\mathbf{x}(\mathrm{E}[A]) = \left[1/6\ 5/24\ 5/8\right]^T .$$

For the random variable model, the computations are more complicated.

$$\begin{aligned} \mathrm{E}[\mathbf{x}(A)] &= \mathrm{E}\left[\sum_{n=0}^{\infty}(A^n \mathbf{P}^{T^n})(1-A)\mathbf{v}\right] \\ &= \sum_{n=0}^{\infty}(\mathrm{E}[A^n] - \mathrm{E}[A^{n+1}])\mathbf{P}^{T^n}\mathbf{v}. \end{aligned} \tag{5}$$

In the first step, we used the Neumann series for the inverse of the matrix $\mathbf{I} - A\mathbf{P}^T$. Fubini's theorem then justifies interchanging the sum and expectation because the inner quantity is always bounded and positive. The raw moments of the uniform distribution are $\mathrm{E}[A^n] = \frac{1}{n+1}$ and consequently,

$$\mathrm{E}[\mathbf{x}(A)] = \sum_{n=0}^{\infty} \left(\tfrac{1}{n+1} - \tfrac{1}{n+2} \right) \mathbf{P}^{T^n} \mathbf{v} = \sum_{n=0}^{\infty} \tfrac{1}{(n+1)(n+2)} \mathbf{P}^{T^n} \mathbf{v}.$$

For $n \geq 2$, $\mathbf{P}^{T^n}\mathbf{v} = \begin{bmatrix} 0 \; 0 \; 1 \end{bmatrix}^T$

$$\begin{aligned} \mathrm{E}[\mathbf{x}(A)] &= \tfrac{1}{2}\mathbf{v} + \tfrac{1}{6}\mathbf{P^T}\mathbf{v} + \left[0 \; 0 \; \textstyle\sum_{n=2}^{\infty} \tfrac{1}{(n+1)(n+2)} \right]^T \\ &= \begin{bmatrix} 1/6 \; 7/36 \; 23/36 \end{bmatrix}^T . \end{aligned}$$

In this case, the first component of the vector is identical, but the second two components show a small change from the modeling difference.

B Engineering Details

Our Matlab implementations, which we provide for download from `http://www.stanford.edu/~dgleich/pagerankpce`, fall into three categories: PageRank codes, polynomial chaos codes, and large scale linear system codes.

PageRank codes. Our PageRank codes use a Gauss-Seidel algorithm [24] to solve the linear system formulation of the PageRank problem. We wrote a mex function to implement the Gauss-Seidel iteration efficiently in a Matlab code. The convergence metric for the PageRank computation was

$$||\alpha \mathbf{P}^T \mathbf{x} + (1 - \alpha)\mathbf{v} - \mathbf{x}||_1 \leq \delta$$

where $\delta = 10^{-10}$ for the PageRank vectors discussed in the results section and $\delta = 10^{-8}$ for the PageRank vectors used to form the Monte Carlo approximation.

Polynomial chaos codes. In the polynomial chaos approach, we must integrate products of the basis polynomials. We computed these quantities exactly with Matlab's symbolic toolbox. We used the output of these symbolic computations when forming the large linear system to solve for the PCE coefficients $\mathbf{x}_j$.

Large scale linear systems. We employed two linear system solvers to compute the solution for the large systems generated by the polynomial chaos approach. For the results in Sect. 6 we solved the final linear systems using the SOR algorithm with $\omega = 1.05$. For the results in Sect. 7, Matlab did not have sufficient memory to construct the large linear system in memory. We represented these matrices implicitly as linear operators and used the Jacobi algorithm to solve the linear systems until the relative residual was smaller than 10^{-10}.