A Deeper Investigation of PageRank as a Function of the Damping Factor*

Paolo Boldi Massimo Santini
Sebastiano Vigna
DSI, Università degli Studi di Milano, Italy

Abstract

PageRank is defined as the stationary state of a Markov chain. The chain is obtained by perturbing the transition matrix induced by a web graph with a damping factor α that spreads uniformly part of the rank. The choice of α is eminently empirical, and in most cases the original suggestion $\alpha = 0.85$ by Brin and Page is still used. In this paper, we give a mathematical analysis of PageRank when α changes. In particular, we show that, contrarily to popular belief, for real-world graphs values of α close to 1 do not give a more meaningful ranking. Then, we give closed-form formulae for PageRank derivatives of any order, and by proving that the k-th iteration of the Power Method gives exactly the PageRank value obtained using a Maclaurin polynomial of degree k, we show how to obtain an approximation of the derivatives. Finally, we view PageRank as a linear operator acting on the preference vector and show a tight connection between iterated computation and derivation.

1 Introduction

PageRank [22] is a ranking technique used by today's search engines. It is query independent and content independent—it can be computed offline using only the web graph[1]. These features make it interesting when we need to assign an absolute measure of importance to each web page.

Originally at the basis of Google's ranking algorithm, PageRank is now just one of the many parameters used by search engines to rank pages. Albeit no public information is available on the current degree of utilisation of PageRank in real-world search engines, it is likely that in certain areas, for instance, selective crawling (deciding which pages to crawl) and inverted index reordering (permuting documents so that more important documents are returned first), PageRank (or one of its many variants) is still very useful. Its influence on the ordering of the results of a query, however, has certainly diminished in time.

Nonetheless, PageRank (and more generally link analysis) is an interesting mathematical phenomenon that has inspired research in a number of fields—for instance, even basic methods commonly used in numerical analysis for matrix computations become tricky to implement when the matrix is of order 10^9. Moreover the matrix induced by a web graph is significantly different from those commonly found in physics or statistics, so many results that are common in those areas are not applicable.

*Part of the results of this paper have been presented at the 14th World–Wide Web Conference [6]. This work has been partially supported by the EC Project DELIS and by the MIUR COFIN projects "Automi e linguaggi formali: aspetti matematici e applicativi" and "Grafi del web e ranking".

[1] The web graph is the directed graph whose nodes are URLs and whose arcs correspond to hyperlinks.

One suggestive way to describe the idea behind PageRank is as follows: consider a random surfer that starts from a random page, and at every time chooses the next page by clicking on one of the links in the current page (selected uniformly at random among the links present in the page). As a first approximation, we could define the rank of a page as the fraction of time that the surfer spent on that page. Clearly, important pages (i.e., pages that happen to be linked by many other pages, or by few important ones) will be visited more often, which justifies the definition.

However, as remarked in [22], this definition would be too simple minded, as certain pages (called therein *rank sinks*, and, in this paper, *buckets*) would end up entrapping the surfer. To solve this problem, at every step we choose a link only with probability α: with probability $1-\alpha$ the surfer will restart instead from another node chosen (uniformly) at random.

A significant part of the current knowledge about PageRank is scattered through the research laboratories of large search engines, and its analysis "has remained largely in the realm of trade secrets and economic competition" [10]. We believe, however, that a scientific and detailed study of PageRank is essential to our understanding of the web (independently of its usage in search engines), and we hope that this paper can be a contribution in such program.

PageRank is defined formally as the stationary distribution of a stochastic process whose states are the nodes of the web graph. The process itself is obtained by mixing the normalised adjacency matrix of the web graph (with some patches for nodes without outlinks that will be discussed later) with a trivial uniform process that is needed to make the mixture irreducible and aperiodic, so that the stationary distribution is well defined. The combination depends on a *damping factor* $\alpha \in [0, 1)$, which will play a major rôle in this paper (and corresponds to the probability that the surfer follows a link of the current page). When α is 0, the web-graph part of the process is annihilated, resulting in the trivial uniform process. As α gets closer to 1, the web part becomes more and more important.

The problem of choosing α was curiously overlooked in the first papers about PageRank: yet, not only PageRank changes significantly when α is modified [24, 23], but also the relative ordering of nodes determined by PageRank can be radically different [20]. The original value suggested by Brin and Page ($\alpha = 0.85$) is the most common choice.

Intuitively, $1-\alpha$ is the fraction of ranking that we agree to spread uniformly on all pages. This amount will be then funneled through the outlinks. A common form of link spamming creates a large set of pages that funnel carefully all their rank towards a single page: even if the set is made of irrelevant pages, they will receive their share of uniformly spread rank, and in the end the page pointed by the set will be given a preposterously great importance.

It is natural to wonder what is the best value of the damping factor, if such a thing exists. In a way, when α gets close to 1 the Markov process is closer to the "ideal" one, which would somehow suggest that α should be chosen as close to 1 as possible. This observation is not new, but there is some naivety in it.

The first issue is of computational nature: PageRank is traditionally computed using variants of the Power Method. The number of iterations required for this method to converge grows with α, and in addition more and more numerical precision is required as α gets closer to 1.

But there is an even more fundamental reason not to choose a value of α too close to 1: we shall prove in Section 5 that when α goes to 1 PageRank gets concentrated in the recurrent states, which correspond essentially to the *buckets*—nondangling nodes whose strongly connected components have no path toward other components. This phenomenon gives a null PageRank to all the pages in the core component, something that is difficult to explain and that is contrary to common sense. In other words, in real-word web graphs the rank of all important nodes (in particular, all nodes of the core component) goes to 0 as α tends to 1.[2]

[2]We remark that in 2006 a very precise analysis of the distribution of PageRank was obtained by [1], corroborating the results described in [6]. Using their analysis, the authors conclude that α should be set equal to $1/2$.

Thus, PageRank starts, when $\alpha = 0$, from a uninformative uniform distribution and ends, when $\alpha \to 1$, into a counterintuitive distribution concentrated mostly in irrelevant nodes. As a result, both for choosing the correct damping factor and for detecting link spamming, being able to describe the behaviour of PageRank when α changes is essential.

To proceed further in this direction, it is essential that we have at our disposal analytical tools that describe this behaviour. To this purpose, we shall provide closed-form formulae for the derivatives of any order of PageRank with respect to α. Moreover, we show that the k-th coefficient of the PageRank power series (in α) can be easily computed during the k-th iteration of the Power Method.

The most surprising consequence, easily derived from our formulae, is that the vectors computed during the PageRank computation for *any* $\alpha \in (0, 1)$ can be used to approximate PageRank for *every other* $\alpha \in (0, 1)$. Of course, the same coefficients can be used to approximate the derivatives, and we provide some simple bound to the precision of the approximation. These observations makes it possible to study easily the behaviour of PageRank for any node when α changes storing a minimal amount of data.[3]

2 Basic definitions

Let G be (the adjacency matrix of) a directed graph of N nodes (identified hereafter with the numbers from 0 to $N - 1$). A node is *terminal* if it does not have outlinks, except possibly for loops (or, equivalently, if all arcs incident on the node are incoming). A *dangling node*[4] is a terminal node without loops.

We note that usually G is preprocessed before building the corresponding Markov chain. Common processing includes removal of all loops (as nodes should not give authoritativeness to themselves) and thresholding the number of links coming from pages of the same domain (to reduce the effect of nepotistic link spamming).

Given a graph G, the *row-normalised matrix* of G is the matrix $\bar{G}$ such that $(\bar{G})_{ij}$ is one over the outdegree of i if there is an arc from i to j in G, zero otherwise.

If no dangling nodes are present, $\bar{G}$ is stochastic and it is the transition matrix of the *natural random walk* on G. Otherwise, rows corresponding to dangling nodes will be entirely made of zeroes and will have to be patched somehow, for instance substituting them with uniform distributions[5]. However, more generally we might substitute rows of zeroes in G with a given fixed distribution, and we shall see that this change has significant consequences on our analysis.

Let us define $\boldsymbol{d}$ as the characteristic vector[6] of dangling nodes (i.e., the vector with 1 in positions corresponding to nodes without outgoing arcs and 0 elsewhere). Let $\boldsymbol{v}$ and $\boldsymbol{u}$ be two distributions, which we will call the *preference* and the *dangling-node* distribution, respectively.

PageRank $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ is defined (up to a scalar) by the eigenvector equation

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)\big(\alpha(\bar{G} + \boldsymbol{d}^T\boldsymbol{u}) + (1-\alpha)\mathbf{1}^T\boldsymbol{v}\big) = \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha),$$

that is, as the stationary state of the Markov chain $\alpha(\bar{G} + \boldsymbol{d}^T\boldsymbol{u}) + (1-\alpha)\mathbf{1}^T\boldsymbol{v}$: such chain is indeed *unichain* [4], so the previous definition is well given. More precisely, we have a *Markov chain with*

[3] Free Java code implementing all the algorithms described in this paper is available for download at `http://law.dsi.unimi.it/`.

[4] The same kind of node is often called a *sink* in graph-theoretic literature. Our choice follows the standard PageRank literature, and avoids the usage of ambiguous terms that have been given different meanings in different papers, in particular w.r.t. the presence of loops.

[5] In this work, by *distribution* we mean a vector with non-negative entries and ℓ_1-norm equal to 1. The indices for which the distribution is non-zero are called its *support*.

[6] All vectors in this work are row vectors.

restart [4] in which $\bar{G} + \boldsymbol{d}^T\boldsymbol{u}$ is the Markov chain (that follows the natural random walk on non-dangling nodes, and moves to a node at random with distribution $\boldsymbol{u}$ when starting from a dangling node) and $\boldsymbol{v}$ is the restart vector. The *damping factor* $\alpha \in [0\,..\,1)$ determines how often the Markov chain follows the graph rather than moving at a random node following the preference vector $\boldsymbol{v}$.

The preference vector is used to bias PageRank w.r.t. a selected set of trusted pages, or might depend on the user's preferences, in which case one speaks of *personalised PageRank* [16]. Clearly, the preference vector conditions significantly PageRank, but in real-world crawls, which have a large number of dangling nodes (in particular if the graph contains the whole *frontier* of the crawl [10], rather than just the visited nodes) the dangling-node distribution $\boldsymbol{u}$ can also be very important. In the literature one can find several alternatives (e.g., $\boldsymbol{u} = \boldsymbol{v}$ or $\boldsymbol{u} = \mathbf{1}/n$).

Following [5], we distinguish clearly between *strongly preferential* PageRank, in which the preference and dangling-node distributions are identical (i.e., $\boldsymbol{u} = \boldsymbol{v}$) and that corresponds to a topic or personalisation bias, and *weakly preferential* PageRank, in which the preference and the dangling-node distributions are not identical, and, in principle, uncorrelated. The distinction is not irrelevant, as the concordance between weakly and strongly preferential PageRank can be quite low [5]. Both strongly and weakly preferential PageRank (and also *pseudoranks* defined in Section 7) have been used in the literature to define PageRank, so a great care must be exercised when comparing results from different papers.

In the rest of the paper, we shall use the following matrices:

$$P_{\boldsymbol{u}} := \bar{G} + \boldsymbol{d}^T\boldsymbol{u}$$
$$M_{\boldsymbol{v},\boldsymbol{u}}(\alpha) := \alpha P_{\boldsymbol{u}} + (1-\alpha)\mathbf{1}^T\boldsymbol{v}$$

As a mnemonic, $P_{\boldsymbol{u}}$ is the *patched* version of $\bar{G}$ in which rows corresponding to dangling nodes have been patched with $\boldsymbol{u}$, and $M_{\boldsymbol{v},\boldsymbol{u}}$ is the actual Markov chain whose stationary distribution is PageRank. Note that, here and elsewhere, when a matrix or a vector is a function of the damping factor $\alpha \in [0, 1)$, we will use a notation that reflects this fact.

We are providing a toy example shown in Figure 1. It will be used in the rest of the paper as a guide.

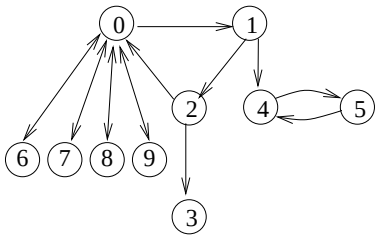

Figure 1: A toy example graph with $N = 10$ nodes.

3 The Many Natures of PageRank

We introduced PageRank as the stationary state of a Markov chain. Actually, due to the presence of the damping factor, PageRank can be seen as a *rational vector function* $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ associating to each value of α a different rank. As α goes from 0 to 1, the ranks change dramatically, and the main theme of this paper is exactly the study of PageRank as a function of the damping factor.

Usually, though, one looks at $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ only for a *specific* value of α. All algorithms to compute PageRank actually compute (or, more precisely, provide an estimate for) $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ for some α that you

plug in it, and it is by now an established use to choose $\alpha = 0.85$. This choice was indeed proposed by Brin and Page [22].

Many authors have tried to devise a more thorough *a posteriori* justification for 0.85. It is easy to get convinced that choosing a small value for α is not appropriate, because too much weight would be given to the "uniform" part of $M_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$. On the other hand, a value of α too close to 1 leads to numerical instability. Using a disciplined approach, based on the assumption that most PageRank mass must belong to the core component, the authors of [1] claim that α should be $1/2$.

Noting that $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)\mathbf{1}^T = 1$, we get

$$\begin{aligned}\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)\big(\alpha P_{\boldsymbol{u}} + (1-\alpha)\mathbf{1}^T\boldsymbol{v}\big) &= \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)\\ \alpha\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} + (1-\alpha)\boldsymbol{v} &= \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)\\ (1-\alpha)\boldsymbol{v} &= \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)(I-\alpha P_{\boldsymbol{u}}),\end{aligned}$$

which yields the following closed formula for PageRank[7]:

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = (1-\alpha)\boldsymbol{v}(I-\alpha P_{\boldsymbol{u}})^{-1}. \tag{1}$$

Note that the above formula exhibits PageRank as a *linear operator* applied to the preference vector $\boldsymbol{v}$. In particular, standard methods for solving linear systems can (and should) be used to compute it much more efficiently than with the Power Method. For instance, since $I - \alpha P_{\boldsymbol{u}}$ is strictly diagonally dominant, the Gauss–Seidel method is guaranteed to converge, and it features in practise faster convergence than the Power Method (see, for example, [9]).

In Figure 2 we show the exact PageRank vector for our toy example, computed using (1).

$$\begin{aligned}\boldsymbol{r}_{\mathbf{1}/10,\mathbf{1}/10}(\alpha) = \Big\langle &\frac{5(1-\alpha)(\alpha^2+18\alpha+4)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, && \frac{2(1-\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200},\\ &\frac{-2(1-\alpha)(7\alpha^2-5\alpha-10)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, && \frac{-(1-\alpha)(11\alpha^2+8\alpha^3-10\alpha-20)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200},\\ &-\frac{\alpha^4+16\alpha^3+14\alpha^2-30\alpha-20}{(\alpha+1)(8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200)}, && -\frac{15\alpha^3+6\alpha^2-20\alpha-20}{(\alpha+1)(8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200)},\\ &\frac{2(1-\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, && \frac{2(1-\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200},\\ &\frac{2(1-\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, && \frac{2(1-\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}\Big\rangle\end{aligned}$$

Figure 2: The explicit formula of PageRank as a function of α with $\boldsymbol{v} = \boldsymbol{u} = \mathbf{1}/10$ for the graph shown in Figure 1.

The linear operator in (1) can be written as

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = (1-\alpha)\boldsymbol{v}\sum_{k=0}^{\infty}(\alpha P_{\boldsymbol{u}})^k, \tag{2}$$

which makes the dependence of PageRank on incoming paths very explicit: PageRank is computed by diffusing the base preference along all outgoing path with decay α. From this formulation it is also immediate to derive the combinatorial description of PageRank of a node x in terms of a summation of weight of paths coming into x [8].

The reader can see the PageRank vector for our worked-out example in Figure 2 (both $\boldsymbol{v}$ and $\boldsymbol{u}$ are set to the uniform vector). PageRank is represented as a function of α in Figure 8.

[7] A particular case of this formula appears in Lemma 3 of [12], albeit the factor $1-\alpha$ is missing, probably due to an oversight.

4 Power series

Equation (2) can be actually rewritten as follows:

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = \boldsymbol{v} + \boldsymbol{v}\sum_{k=1}^{\infty}\alpha^k\left(P_{\boldsymbol{u}}^k - P_{\boldsymbol{u}}^{k-1}\right). \tag{3}$$

This formula suggests a way to study PageRank as a power series of α. If we want to follow this route, we must overcome two difficulties: first of all, we must compute explicitly the coefficients of the power series[8]; and then, we must discuss how good is the approximation obtained by truncating the series at a given step. Both problems will be solved by a surprisingly simple relationship between the power series and the Power Method that will be proved in this section. To obtain our main result, we will need the following lemma (that can be easily restated in any **R**-algebra):

Lemma 1 *Let $\mathscr{C}$ be a set of square matrices of the same size, and $Z \in \mathscr{C}$ such that for every $A \in \mathscr{C}$ we have $AZ = Z$. Then for all $A \in \mathscr{C}$, $\lambda \in \mathbf{R}$ and for all n we have*

$$(\lambda A + (1-\lambda)Z)^n = \lambda^n A^n + (1-\lambda)\sum_{k=0}^{n-1}\lambda^k Z A^k$$

or, equivalently,

$$(\lambda A + (1-\lambda)Z)^n = (I - Z)\lambda^n A^n + Z\Big(I + \sum_{k=1}^{n}\lambda^k\left(A^k - A^{k-1}\right)\Big).$$

Proof. By an easy induction. The first statement is trivial for $n = 0$. If we multiply both members by $\lambda A + (1-\lambda)Z$ on the right we have

$$\begin{aligned}
&(\lambda A + (1-\lambda)Z)^{n+1} = \\
&\lambda^{n+1}A^{n+1} + (1-\lambda)\sum_{k=0}^{n-1}\lambda^{k+1}ZA^{k+1} + \lambda^n(1-\lambda)Z + (1-\lambda)^2\sum_{k=0}^{n-1}\lambda^k Z = \\
&= \lambda^{n+1}A^{n+1} + (1-\lambda)\sum_{k=0}^{n-1}\lambda^{k+1}ZA^{k+1} + \lambda^n(1-\lambda)Z + (1-\lambda)^2\frac{1-\lambda^n}{1-\lambda}Z = \\
&= \lambda^{n+1}A^{n+1} + (1-\lambda)\sum_{k=0}^{n}\lambda^k ZA^k.
\end{aligned}$$

The second statement can be then proved by expanding the summation and collecting monomials according to the powers of λ. ∎

We are now ready for the main result of this section, which equates analytic approximation (the index at which we truncate the PageRank power series) with computational approximation (the number of iterations of the Power Method):

Theorem 1 *The approximation of PageRank computed at the n-th iteration of the Power Method with damping factor α and starting vector $\boldsymbol{v}$ coincides with the n-th degree truncation of the power series of PageRank evaluated in α. In other words,*

$$\boldsymbol{v}M_{\boldsymbol{v},\boldsymbol{u}}^n = \boldsymbol{v} + \boldsymbol{v}\sum_{k=1}^{n}\alpha^k\left(P_{\boldsymbol{u}}^k - P_{\boldsymbol{u}}^{k-1}\right).$$

[8] Note that the coefficients are vectors, because we are approximating a vector function.

	Coefficient
α^0	$\langle 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100\rangle$
α^1	$\langle 0.371, -0.058, -0.028, -0.028, 0.015, -0.034, -0.058, -0.058, -0.058, -0.058\rangle$
α^2	$\langle -0.253, 0.070, -0.033, -0.018, -0.048, 0.003, 0.070, 0.070, 0.070, 0.070\rangle$
α^3	$\langle 0.260, -0.055, 0.030, -0.021, 0.032, -0.026, -0.055, -0.055, -0.055, -0.055\rangle$
α^4	$\langle -0.207, 0.050, -0.029, 0.013, -0.040, 0.012, 0.050, 0.050, 0.050, 0.050\rangle$

Table 1: The coefficients of the first terms of the power series for $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$.

Proof. Apply Lemma 1 to the case when $A = P_{\boldsymbol{u}}$, $Z = \mathbf{1}^T\boldsymbol{v}$ and $\lambda = \alpha$. We have:

$$M^n_{\boldsymbol{v},\boldsymbol{u}} = \big(\alpha P_{\boldsymbol{u}} + (1-\alpha)\mathbf{1}^T\boldsymbol{v}\big)^n = (I - \mathbf{1}^T\boldsymbol{v})\lambda^n P^n_{\boldsymbol{u}} + \mathbf{1}^T\boldsymbol{v}\Big(I + \sum_{k=1}^{n}\lambda^k\big(P^k_{\boldsymbol{u}} - P^{k-1}_{\boldsymbol{u}}\big)\Big)$$

hence, noting that $\boldsymbol{v}\mathbf{1}^T\boldsymbol{v} = \boldsymbol{v}$,

$$\boldsymbol{v}M^n_{\boldsymbol{v},\boldsymbol{u}} = \boldsymbol{v}(I - \mathbf{1}^T\boldsymbol{v})\lambda^n P^n_{\boldsymbol{u}} + \boldsymbol{v}\mathbf{1}^T\boldsymbol{v}\Big(I + \sum_{k=1}^{n}\lambda^k\big(P^k_{\boldsymbol{u}} - P^{k-1}_{\boldsymbol{u}}\big)\Big) = 0 + \boldsymbol{v} + \boldsymbol{v}\sum_{k=1}^{n}\lambda^k\big(P^k_{\boldsymbol{u}} - P^{k-1}_{\boldsymbol{u}}\big). \blacksquare$$

As a consequence:

Corollary 1 *The difference between the k-th and the $(k-1)$-th approximation of PageRank (as computed by the Power Method with starting vector $\boldsymbol{v}$), divided by α^k, is the k-th coefficient of the power series of PageRank.*

The previous corollary is apparently innocuous; however, it has a surprising consequence: the data obtained computing PageRank for a given α, say[9] α_0, can be used to compute immediately PageRank for *any other* α_1, obtaining the same result that we would have obtained after the same number of iterations of the Power Method with $\alpha = \alpha_1$. Indeed, by saving the coefficients of the power series during the computation of PageRank with a specific α it is possible to study the behaviour of PageRank when α varies. Even more is true, of course: using standard series derivation techniques, one can approximate the k-th derivative. A useful bound for approximating derivatives will be given in Section 6.2.

The first few coefficients of the power series for our worked-out example are shown in Table 1. Figure 3 shows the convergence of the power series toward the actual PageRank behaviour for a chosen node. Finally, in Figure 4 we display the approximation obtained with truncating the power series after the first 100 terms. We choose four nodes with different behaviours (monotonic increasing/decreasing, unimodal concave/convex) to show that the approximation is excellent in all these cases. For this experiment we used a 41 291 594-nodes snapshot of the Italian web gathered by UbiCrawler [3] and indexed by WebGraph [7].

5 Limit behaviour

Power series offer an easy numerical way to study the behaviour of PageRank as a function of α, but as α gets closer to 1 the approximation needs more and more terms to be useful (the other extremal behaviour, i.e., $\alpha = 0$, is trivial, since $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(0) = \boldsymbol{v}$). Thus, this section is devoted to a formal analysis of the behaviour of PageRank when α is in a neighbourhood of 1.

[9] Actually, to compute the coefficients one can even use $\alpha_0 = 1$.

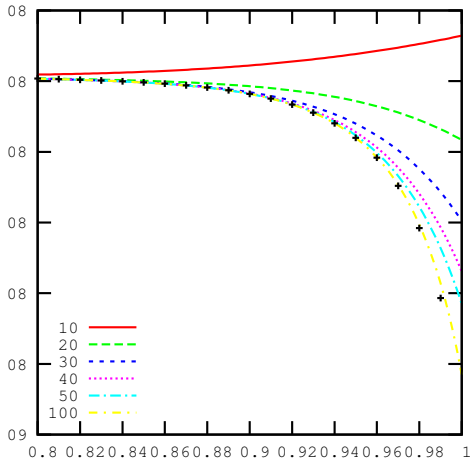

Figure 3: Approximating $\boldsymbol{r}(\alpha)$ for a specific node (cross-shaped points) using Maclaurin polynomials of different degrees (shown in the legend).

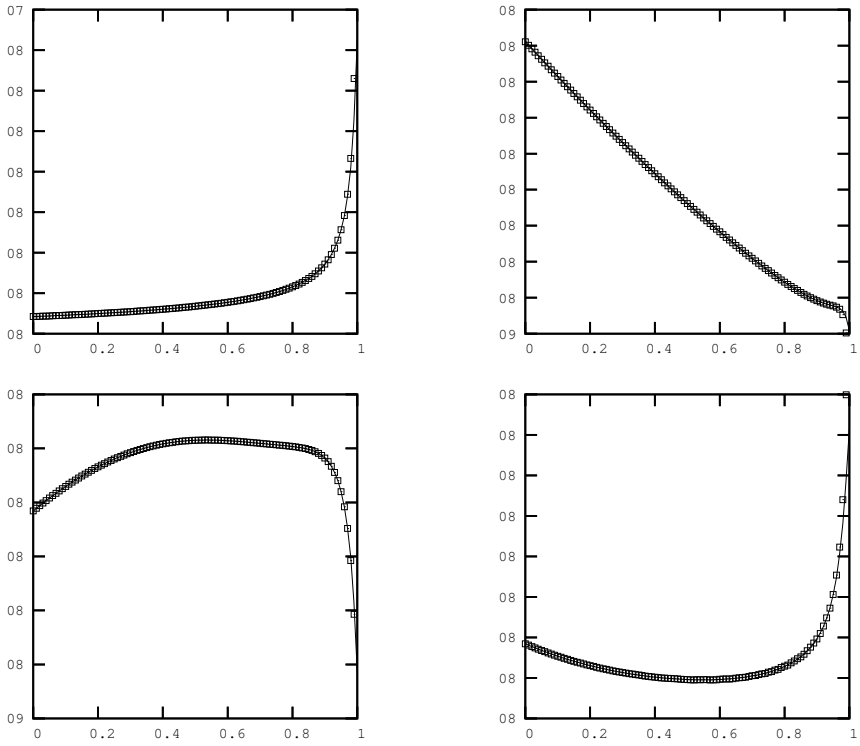

Figure 4: Examples of approximations obtained using a Maclaurin polynomial of degree 100, for nodes with different behaviours (the points were tabulated by computing PageRank explicitly with 100 regularly spaced values of α).

When $\alpha \to 1^-$, the transition matrix $M_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ tends to $P_{\boldsymbol{u}}$: this fact seems to suggest that choosing α close to 1 should give a "truer" or "better" PageRank: this is a widely diffused opinion (as we shall see, most probably a misconception). In any case, as we remarked in the introduction there are some computational obstacles to choosing a value of α too close to 1. The Power Method converges more and more slowly [14] as $\alpha \to 1^-$, a fact that also influences the other methods used to compute PageRank (which are often themselves variants of the Power Method [22, 13, 11, 21, 18, 17]). Indeed, the number of iterations required could in general be bounded using the separation between the first and the second eigenvalue, but unfortunately the separation can be abysmally small if $\alpha = 1$, making this technique not applicable. Moreover, if α is large the computation of PageRank may become numerically ill-conditioned (essentially for the same reason [12]).

Even disregarding the problems discussed above, we shall provide convincing reasons that make it inadvisable to use a value of α close to 1, unless $P_{\boldsymbol{u}}$ is suitably modified. First observe that, since $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ is a rational (coordinatewise) bounded function defined on $[0, 1)$, it is defined on the whole complex plane except for a finite number of poles, and the limit

$$\boldsymbol{r}^*_{\boldsymbol{v},\boldsymbol{u}} = \lim_{\alpha \to 1^-} \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$$

exists. In fact, since the *resolvent* $I/\alpha - P_{\boldsymbol{u}}$ has a Laurent expansion[10] around 1 in the largest disc not containing $1/\lambda$ for another eigenvalue λ of $P_{\boldsymbol{u}}$, PageRank is analytic in the same disc; a standard computation yields

$$(1-\alpha)(1-\alpha P_{\boldsymbol{u}})^{-1} = P^*_{\boldsymbol{u}} - \sum_{n=0}^{\infty} \left(\frac{\alpha-1}{\alpha}\right)^{n+1} Q^{n+1}_{\boldsymbol{u}},$$

where $Q_{\boldsymbol{u}} = (I - P_{\boldsymbol{u}} + P^*_{\boldsymbol{u}})^{-1} - P^*_{\boldsymbol{u}}$ and

$$P^*_{\boldsymbol{u}} = \lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} P^k_{\boldsymbol{u}}$$

is the *Cesáro limit* of $P_{\boldsymbol{u}}$ [15], hence

$$\boldsymbol{r}^*_{\boldsymbol{v},\boldsymbol{u}} = \boldsymbol{v} P^*_{\boldsymbol{u}}.$$

Figure 5 exhibits PageRank of a node as a complex function.

It is easy to see that $\boldsymbol{r}^*_{\boldsymbol{v},\boldsymbol{u}}$ is actually *one of the* invariant distributions of $P_{\boldsymbol{u}}$ (because $\lim_{\alpha \to 1^-} M_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = P_{\boldsymbol{u}}$). Can we somehow characterise the properties of $\boldsymbol{r}^*_{\boldsymbol{v},\boldsymbol{u}}$? And what makes $\boldsymbol{r}^*_{\boldsymbol{v},\boldsymbol{u}}$ different from the other (infinitely many, if $P_{\boldsymbol{u}}$ is reducible) invariant distributions of $P_{\boldsymbol{u}}$?

The first question is the most interesting, because it is about what happens to PageRank when $\alpha \to 1^-$; in a sense, fortunately, it is also the easiest to answer. Before doing this, recall some basic definitions and facts about Markov chains.

- Given two states x and y, we say that x *leads to* y iff there is some $m > 0$ such that there is a non-zero probability to go from x to y in m steps.
- A state x is *transient* iff there is a state y such that x leads to y but y does not lead to x. A state is *recurrent* iff it is not transient.
- In every invariant distribution $\boldsymbol{p}$ of an aperiodic Markov chain, if $p_x > 0$ then x is recurrent [15].

[10] A different expansion around 1, based on vector-extrapolation techniques, has been proposed in [25].

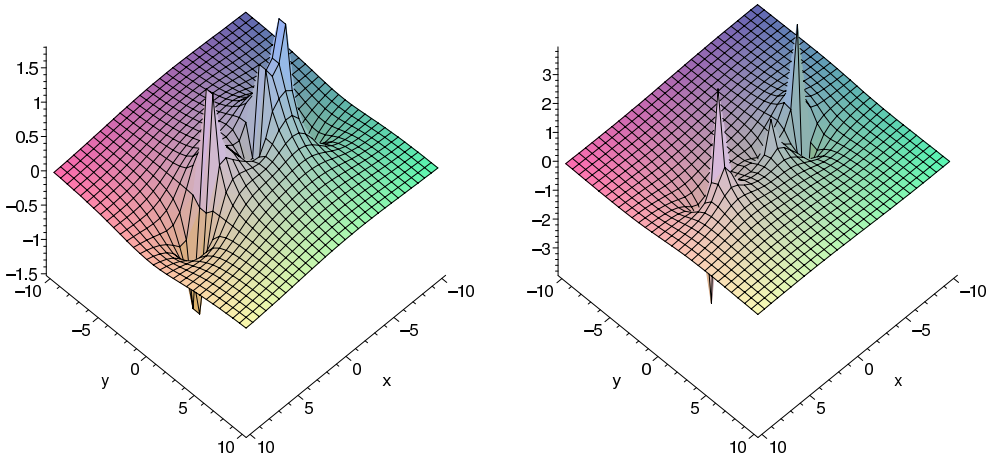

Figure 5: The real and imaginary parts of PageRank of node 0 of the graph shown in Figure 1, plotted for all complex values with real and imaginary parts smaller than 10. Poles appear as spikes.

Let us now introduce some graph-theoretical notation. Let G be a graph.

- Given a node x of G, we write $[x]_G$ for the (strongly connected) component of G containing x.
- The *component graph* of G is a graph whose nodes are the components of G, with an arc from $[x]_G$ to $[y]_G$ iff there are nodes $x' \in [x]_G$ and $y' \in [y]_G$ such that there is an arc from x' to y' in G. The component graph is acyclic, apart for the possible presence of loops.
- If x, y are two nodes of G, we write $x \rightsquigarrow_G y$ iff there is a directed (possibly empty) path from x to y in G.

Clearly, a node is recurrent in $P_{\boldsymbol{u}}$ iff $[x]_{P_{\boldsymbol{u}}}$ is terminal; otherwise said, x is recurrent (in the Markov chain $P_{\boldsymbol{u}}$) iff $x \rightsquigarrow_{P_{\boldsymbol{u}}} y$ implies $y \rightsquigarrow_{P_{\boldsymbol{u}}} x$ as well. Note that nodes with just a loop *are* recurrent.

We now turn to our characterisation theorem, which identifies recurrent states on the basis of G, rather than $P_{\boldsymbol{u}}$. The essence of the theorem is that, for what concerns recurrent states, the difference between G and $P_{\boldsymbol{u}}$ is not significant, except for a special case which, however, is as pathological as periodicity in a large web graph.

To state and prove comfortably the next theorem, we need a definition:

Definition 1 *A component is said to be a* bucket component *if it is terminal in the component graph, but it is not dangling (e.g., if it contains at least one arc, or, equivalently, if the component has a loop in the component graph). A* bucket (node) *is a node belonging to a bucket component.*

Note that given a component $[x]$ of a graph, it is always possible to reach a terminal component starting from $[x]$; such a component must be either dangling or a bucket. We shall use this fact tacitly in the following proof.

Theorem 2 *Let G and $P_{\boldsymbol{u}}$ be defined as above. Then:*

1. *if at least one bucket of G is reachable from the support of $\boldsymbol{u}$ then a node is recurrent for $P_{\boldsymbol{u}}$ iff it is a bucket of G; hence, given an invariant distribution $\boldsymbol{p}$ for $P_{\boldsymbol{u}}$, $p_x > 0$ implies that x is a bucket of G;*

2. *if no bucket of G is reachable from the support of $\boldsymbol{u}$, all nodes reachable from the support of $\boldsymbol{u}$ form a bucket component of $P_{\boldsymbol{u}}$; hence, a node is recurrent for $P_{\boldsymbol{u}}$ iff it is in a bucket component of G or it is reachable from the support of $\boldsymbol{u}$.*

Proof. First of all, *buckets of G are always recurrent for $P_{\boldsymbol{u}}$*. If x is a bucket of G and $x \rightsquigarrow_{P_{\boldsymbol{u}}} y$, a path from x to y cannot traverse a dangling node of G (because x is a bucket), so actually $x \rightsquigarrow_G y$, which implies that y is in the same component as x, so $y \rightsquigarrow_G x$ as well (and $y \rightsquigarrow_{P_{\boldsymbol{u}}} x$ *a fortiori*).

For the converse of (1), suppose that a bucket is reachable from the support of $\boldsymbol{u}$, let x be a non-bucket node and consider any terminal y such that $x \rightsquigarrow_G y$. We distinguish two cases:

- if y is a bucket, $y \rightsquigarrow_{P_{\boldsymbol{u}}} x$ does not hold (from y you can only reach nodes of $[y]_G$ both in G and in $P_{\boldsymbol{u}}$), so x is not recurrent;
- otherwise, if y is dangling (note that it might happen that $x = y$); but by hypothesis we can go in $P_{\boldsymbol{u}}$ from y to a node in the support of $\boldsymbol{u}$ that reaches in G (and *a fortiori* in $P_{\boldsymbol{u}}$) a bucket z, so $x \rightsquigarrow_{P_{\boldsymbol{u}}} z$, but z can only reach nodes in $[z]_G$ both in G and in $P_{\boldsymbol{u}}$; thus, also in this case $z \rightsquigarrow_{P_{\boldsymbol{u}}} x$ does not hold, and x is not recurrent.

For case (2), take two nodes x and y of G reachable from the support of $\boldsymbol{u}$. There are two dangling nodes x' and y' such that $x \rightsquigarrow_G x'$ and $y \rightsquigarrow_G y'$. Since $x' \rightsquigarrow_{P_{\boldsymbol{u}}} y$ and $y' \rightsquigarrow_{P_{\boldsymbol{u}}} x$ (being x and y in the support of $\boldsymbol{u}$), we conclude that x and y are in the same component of $P_{\boldsymbol{u}}$, which is necessarily a bucket, so they are both recurrent.

If, on the other hand, x is not reachable from the support of $\boldsymbol{u}$ and is not a bucket, take as above a dangling node x' reachable from x and a node y in the support of $\boldsymbol{u}$: we then have $x \rightsquigarrow_G x' \rightsquigarrow_{P_{\boldsymbol{u}}} y$ but not $y \rightsquigarrow_{P_{\boldsymbol{u}}} x$, so x is not recurrent. ∎

For the standard weakly preferential assumption $\boldsymbol{u} = \mathbf{1}/n$, and indeed whenever the vector $\boldsymbol{u}$ is strictly positive, the statement can be quite simplified:

Corollary 2 *Assume that $\boldsymbol{u} > 0$ (i.e., $u_x > 0$ for every x). Then:*

1. *if G contains a bucket then a node is recurrent for $P_{\boldsymbol{u}}$ iff it is a bucket of G;*
2. *if G does not contain a bucket all nodes are recurrent for $P_{\boldsymbol{u}}$.*

The statement of the previous theorem may seem a bit unfathomable. The essence, however, could be stated as follows: except for extremely pathological cases (graphs whose only terminal components are dangling nodes, or, more generally, graphs with no bucket reachable from the support of $\boldsymbol{u}$), the recurrent nodes are exactly the buckets. Buckets are often called *rank sinks*, as they absorb all the rank circulating through the graph, but we prefer to avoid the term "sink" as it is already quite overloaded in the graph-theoretical literature. To help the reader understand Theorem 2, we show a pictorial example in Figure 6.

As we remarked, a real-world graph will certainly contain at least one bucket reachable from $\boldsymbol{u}$, so the first statement of the theorem will hold. This means that *most* nodes x will be such that $(\boldsymbol{r}^*_{\boldsymbol{v},\boldsymbol{u}})_x = 0$. In particular, this will be true of all the nodes in the core component [19]: this result is somehow surprising, because it means that many important Web pages (that are contained in the core component) will have rank 0 in the limit (see, for instance, node 0 in our worked-out example). A detailed analysis of this limit behaviour can be found in [1].

This is a rather convincing justification that, contradicting common beliefs, choosing α too close to 1 *does not* provide any good PageRank. Rather, PageRank becomes "sensible" somewhere in between 0 and 1. If we are interested in studying PageRank-like phenomena in the neighbourhood of 1, PageRank variants such as TruRank [26] should be used instead.

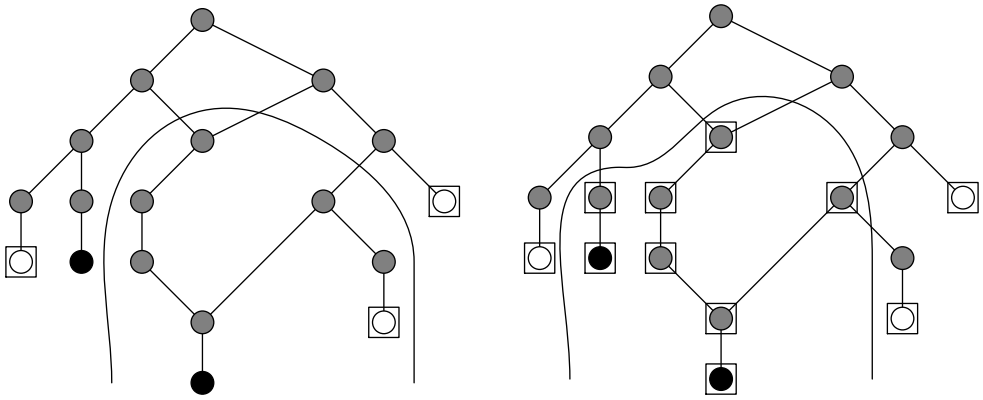

Figure 6: Illustration of Theorem 2. The picture represents the component DAG of a graph (gray=non-terminal component; black=dangling component; white=bucket component); the curve indicates the part of the graph reachable from the support of $\boldsymbol{u}$, and squared components indicate recurrent components. (Left) A situation covered by Theorem 2(1). (Right) The pathological situation covered by Theorem 2(2).

To clarify the above discussion, let us apply it to our toy example (always assuming $\boldsymbol{u} = \boldsymbol{v} = \mathbf{1}/10$). Node 3 is the only dangling node of the graph, but nodes 4 and 5 form a bucket component; all the other nodes are actually in a unique non-terminal component. Thus, the nonzero elements of $P_{\boldsymbol{u}}$ correspond exactly to the arcs of G and to the arcs connecting node 3 to every node in the graph, as shown in Figure 7 (left), where dotted arcs are those that were not present in G. Figure 7 (right) represents the component graph, and the dotted area encloses the components that are actually merged together by patching the dangling node. We are in the conditions of the first item of Corollary 2, and correspondingly Figure 8 shows that PageRank for nodes 4 and 5 grows, whereas for all other nodes it goes to 0 as $\alpha \to 1^-$. Note, however, the maximum attained by node 0 at $\alpha \approx 0.7$.

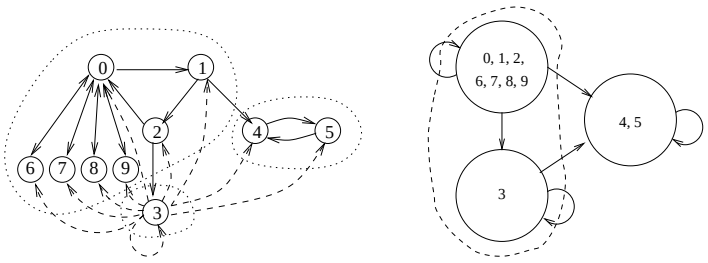

Figure 7: The components of the graph in Figure 1 after the only dangling node has been patched with uniform distribution $\boldsymbol{u} = \mathbf{1}/10$ (the arcs induced by the patching process are dashed) and the corresponding component graph. The dashed line in the component graph gathers components that are merged by the patching process. The only bucket component is $\{4, 5\}$.

6 Derivatives

The reader should by now be convinced that the behaviour of PageRank with respect to the damping factor is nonobvious: $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ should be considered a function of α, and studied as such.

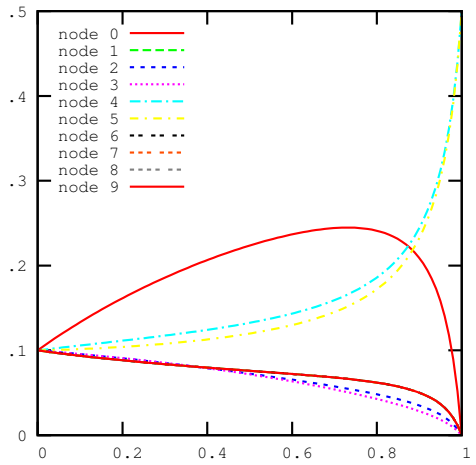

Figure 8: The behaviour of the components of $\boldsymbol{r}_{\mathbf{1}/10,\mathbf{1}/10}(\alpha)$. They all go to zero except for nodes 4 and 5—the only nodes belonging to a bucket component. Note, however, the maximum attained by node 0 at $\alpha \approx 0.7$.

The standard tool for understanding changes in a real-valued function is the analysis of its derivatives. Correspondingly, we are going to provide mathematical support for this analysis.

6.1 Exact formulae

The main objective of this section is providing exact formulae for the derivatives of PageRank. Define $\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$, $\boldsymbol{r}''_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$, ..., $\boldsymbol{r}^{(k)}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ as the first, second, ..., k-th derivative of $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ with respect to α. We start by providing the basic relations between these vector functions:

Theorem 3 *The following identities hold:*

1. $\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = (\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} - \boldsymbol{v})(I - \alpha P_{\boldsymbol{u}})^{-1}$;
2. *for all* $k > 0$, $\boldsymbol{r}^{(k+1)}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = (k+1)\boldsymbol{r}^{(k)}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}}(I - \alpha P_{\boldsymbol{u}})^{-1}$.

Proof. Multiplying both sides of (1) by $I - \alpha P_{\boldsymbol{u}}$ and differentiating memberwise:

$$\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)(I - \alpha P_{\boldsymbol{u}}) - \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} = -\boldsymbol{v} \tag{4}$$

$$\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)(I - \alpha P_{\boldsymbol{u}}) = \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} - \boldsymbol{v} \tag{5}$$

$$\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = (\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} - \boldsymbol{v})(I - \alpha P_{\boldsymbol{u}})^{-1}. \tag{6}$$

This proves the first item; multiplying again both sides by $I - \alpha P_{\boldsymbol{u}}$ and differentiating memberwise we obtain:

$$\begin{aligned}
\boldsymbol{r}''_{\boldsymbol{v},\boldsymbol{u}}(\alpha)(I - \alpha P_{\boldsymbol{u}}) - \boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} &= \boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} \\
\boldsymbol{r}''_{\boldsymbol{v},\boldsymbol{u}}(\alpha)(I - \alpha P_{\boldsymbol{u}}) &= 2\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}} \\
\boldsymbol{r}''_{\boldsymbol{v},\boldsymbol{u}}(\alpha) &= 2\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)P_{\boldsymbol{u}}(I - \alpha P_{\boldsymbol{u}})^{-1}.
\end{aligned}$$

which accounts for the base case ($k = 1$) of an induction for the second statement. For the inductive step, multiplying both sides of the inductive hypothesis by $I - \alpha P_{\boldsymbol{u}}$ and differentiating memberwise:

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+2)}(\alpha)(I - \alpha P_{\boldsymbol{u}}) - \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+1)}(\alpha) P_{\boldsymbol{u}} &= (k+1)\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+1)}(\alpha) P_{\boldsymbol{u}} \\ \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+2)}(\alpha)(I - \alpha P_{\boldsymbol{u}}) &= (k+2)\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+1)}(\alpha) P_{\boldsymbol{u}} \\ \boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+2)}(\alpha) &= (k+2)\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k+1)}(\alpha) P_{\boldsymbol{u}}(I - \alpha P_{\boldsymbol{u}})^{-1} \end{aligned}$$

which accounts for the inductive step. ∎

Moreover, we can explicitly write a closed formula for the generic derivative:

Corollary 3 *For every $k > 0$*

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}^{(k)}(\alpha) = k!\boldsymbol{v}\big(P_{\boldsymbol{u}}^{k} - P_{\boldsymbol{u}}^{k-1}\big)(I - \alpha P_{\boldsymbol{u}})^{-(k+1)}.$$

Proof. The formula can be verified by induction on k, using Theorem 3 and (1). ∎

6.2 Approximating the derivatives

The formulae obtained in Section 6 do not lead directly to an effective algorithm that computes derivatives: even assuming that the exact value of $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ is available, to obtain the derivatives one should invert $I - \alpha P_{\boldsymbol{u}}$ (see Theorem 3), a heavy (in fact, unfeasible) computational task. However, in this section we shall provide a way to obtain simultaneous approximations for PageRank and its derivatives, and we will show how these approximations converge to the desired vectors. The technique we describe is essentially an extension of the Power Method that infers values of the derivatives by exploiting the connection pointed out in Theorem 1.

First of all, note that the k-derivative can be obtained by deriving formally (2). To simplify the notation in the following computations, we rewrite (2) with a more compact notation:

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = \boldsymbol{v} + \boldsymbol{v}\sum_{n=1}^{\infty}\alpha^n\big(P_{\boldsymbol{u}}^{n} - P_{\boldsymbol{u}}^{n-1}\big) = \sum_{n=0}^{\infty}\boldsymbol{a}_n\alpha^n,$$

where $\boldsymbol{a}_0 = \boldsymbol{v}$ and, for $n > 0$, $\boldsymbol{a}_n = \boldsymbol{v}\big(P_{\boldsymbol{u}}^{n} - P_{\boldsymbol{u}}^{n-1}\big)$. By formal derivation, we obtain

$$\boldsymbol{r}^{(k)}(\alpha) = \sum_{n=0}^{\infty} n^{\underline{k}}\,\boldsymbol{a}_n\alpha^n, \tag{7}$$

where we dropped the dependency on $\boldsymbol{v}$ and $\boldsymbol{u}$ to make notation less cluttered, and $n^{\underline{k}}$ denotes the *falling factorial* $n^{\underline{k}} = n(n-1)(n-2)\cdots(n-k+1)$. The Maclaurin polynomials of order t (that is, the t-th partial sum of the series (7)) will be denoted by $[\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t$.

Theorem 4 *If $t \geq k/(1-\alpha)$,*

$$\big\|\boldsymbol{r}^{(k)}(\alpha) - [\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t\big\| \leq \frac{\delta_t}{1-\delta_t}\big\|[\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t - [\![\boldsymbol{r}^{(k)}(\alpha)]\!]_{t-1}\big\|,$$

where

$$1 > \delta_t = \frac{\alpha(t+1)}{t+1-k}.$$

Note that $\alpha \leq \delta_t < 1$ and that $\delta_t \to \alpha$ monotonically, so the theorem states that the error at step t is ultimately bounded by the difference between the t-th and the $(t-1)$-th approximation. The difference between the two approximations is actually the t-th term, so we can also write

$$\left\| \boldsymbol{r}^{(k)}(\alpha) - [\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t \right\| \leq \frac{\delta_t}{1-\delta_t} \alpha^t t^{\underline{k}} \|\boldsymbol{a}_t\|.$$

As a corollary,

Corollary 4 $\left\| \boldsymbol{r}^{(k)}(\alpha) - [\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t \right\| = O(t^k \alpha^t)$.

Note, however, that in practice Theorem 4 is much more useful than the last corollary, as convergence is usually quicker than $O(t^k \alpha^t)$ (much like the actual, error-estimated convergence of the Power Method for the computation of PageRank is quicker than the trivial $O(\alpha^t)$ bound would imply).

Proof (of Theorem 4). We have to bound

$$\left\| \boldsymbol{r}^{(k)}(\alpha) - [\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t \right\| = \left\| \sum_{n=t+1}^{\infty} n^{\underline{k}} \boldsymbol{a}_n \alpha^n \right\|.$$

Since

$$\left\| \boldsymbol{v}(P_{\boldsymbol{u}}^{n+1} - P_{\boldsymbol{u}}^{n}) \right\| \leq \left\| \boldsymbol{v}(P_{\boldsymbol{u}}^{n} - P_{\boldsymbol{u}}^{n-1}) \right\| \|P_{\boldsymbol{u}}\|, \qquad \text{and} \qquad (n+1)^{\underline{k}} \alpha^{n+1} = \frac{\alpha(n+1)}{n+1-k} n^{\underline{k}} \alpha^n$$

the terms of the power series obey the following upper bound:

$$(n+1)^{\underline{k}} \alpha^{n+1} \|\boldsymbol{a}_{n+1}\| \leq \frac{\alpha(n+1)}{n+1-k} n^{\underline{k}} \alpha^n \|\boldsymbol{a}_n\|.$$

Thus, for every $n \geq t \geq 0$ we have the bound

$$n^{\underline{k}} \alpha^n \|\boldsymbol{a}_n\| \leq \left(\frac{\alpha(t+1)}{t+1-k} \right)^{n-t} t^{\underline{k}} \alpha^t \|\boldsymbol{a}_t\| = \delta_t^{n-t} t^{\underline{k}} \alpha^t \|\boldsymbol{a}_t\|.$$

Hence, if t is such that $\delta_t < 1$, we have

$$\left\| \boldsymbol{r}^{(k)}(\alpha) - [\![\boldsymbol{r}^{(k)}(\alpha)]\!]_t \right\| \leq \sum_{n=t+1}^{\infty} \delta_t^{n-t} t^{\underline{k}} \|\boldsymbol{a}_t\| \alpha^t = \frac{\delta_t}{1-\delta_t} t^{\underline{k}} \|\boldsymbol{a}_t\| \alpha^t. \blacksquare$$

The above results suggest a very simple way to compute any desired set of derivatives. Just run the Power Method and, as suggested in Section 4, gather the coefficients of the PageRank power series. Multiplying the n-th coefficient by $n^{\underline{k}}$ is sufficient to get the coefficient for the k-derivative, and after $k/(1-\alpha)$ steps it will be possible to estimate the convergence using (4).

The same considerations made before apply: by storing the coefficients of the Maclaurin polynomials it will be possible to approximate *every* derivative for *every* value of α, albeit the approximation will be worse as the derivative index raises and as $\alpha \to 1$.

7 PageRank as a function of the preference vector

The dependence of PageRank on the preference vector $\boldsymbol{v}$ and on the dangling-node distribution $\boldsymbol{u}$ is also a topic that deserves some attention. With this aim, let us define the *pseudorank* [5] (in G) of a distribution $\boldsymbol{x}$ and damping factor $\alpha \in [0\,..\,1)$ as

$$\widetilde{\boldsymbol{x}}(\alpha) = (1-\alpha)\boldsymbol{x}\big(I - \alpha\bar{G}\big)^{-1}.$$

For every fixed α, the pseudorank is a linear operator and the above definition can be extended by continuity to $\alpha = 1$ even when 1 is an eigenvalue of $\bar{G}$, always using the fact that $I/\alpha - \bar{G}$ has a Laurent expansion around 1; once more,

$$\lim_{\alpha\to 1^-} \widetilde{\boldsymbol{x}}(\alpha) = \boldsymbol{x}\bar{G}^*.$$

When $\alpha < 1$ the matrix $I - \alpha\bar{G}$ is strictly diagonally dominant, so the Gauss–Seidel method can still be used to compute pseudoranks efficiently.

Armed with this definition, we state the main result of [5] (an application of the Sherman–Morrison formula to equation (2)):

$$\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = \widetilde{\boldsymbol{v}}(\alpha) - \widetilde{\boldsymbol{u}}(\alpha)\frac{\boldsymbol{d}\widetilde{\boldsymbol{v}}(\alpha)^T}{1-\frac{1}{\alpha}+\boldsymbol{d}\widetilde{\boldsymbol{u}}(\alpha)^T}. \tag{8}$$

The above formula makes the dependence on the preference and dangling-node distributions very explicit.

In particular, we notice that the dependence on the dangling-node distribution is *not linear*, so *we cannot expect strongly preferential PageRank to be linear in* $\boldsymbol{v}$, because in that case $\boldsymbol{v}$ is also used as dangling-node distribution. Nonetheless, once the pseudoranks for certain distributions have been computed, the above formula makes it possible to compute PageRank using any *convex combination* of such distributions as preference and dangling-node distribution.

However, if we let $\boldsymbol{u} = \boldsymbol{v}$ in (8) (getting back the formula obtained by Del Corso, Gullì and Romani [9])[11], we obtain

$$\boldsymbol{r}_{\boldsymbol{v}}(\alpha) = \widetilde{\boldsymbol{v}}(\alpha)\left(1 - \frac{\boldsymbol{d}\widetilde{\boldsymbol{v}}(\alpha)^T}{1-\frac{1}{\alpha}+\boldsymbol{d}\widetilde{\boldsymbol{v}}(\alpha)^T}\right), \tag{9}$$

where we used $\boldsymbol{r}_{\boldsymbol{v}}(\alpha)$ in place of $\boldsymbol{r}_{\boldsymbol{v},\boldsymbol{v}}(\alpha)$ for brevity. As observed in [9], this formula shows that the strongly preferential PageRank with preference vector $\boldsymbol{v}$ is actually equal, up to normalization, to the pseudorank of $\boldsymbol{v}$. Hence, in particular, even though strongly preferential PageRank is not linear, if $\boldsymbol{v} = \lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}$, then

$$\boldsymbol{r}_{\boldsymbol{v}}(\alpha) = \widetilde{\boldsymbol{v}}(\alpha)\left(1 - \frac{\boldsymbol{d}^T\widetilde{\boldsymbol{v}}(\alpha)}{1-\frac{1}{\alpha}+\boldsymbol{d}^T\widetilde{\boldsymbol{v}}(\alpha)}\right) =$$
$$\lambda\widetilde{\boldsymbol{x}}(\alpha)\left(1 - \frac{\boldsymbol{d}^T\widetilde{\boldsymbol{v}}(\alpha)}{1-\frac{1}{\alpha}+\boldsymbol{d}^T\widetilde{\boldsymbol{v}}(\alpha)}\right) + (1-\lambda)\widetilde{\boldsymbol{y}}(\alpha)\left(1 - \frac{\boldsymbol{d}^T\widetilde{\boldsymbol{v}}(\alpha)}{1-\frac{1}{\alpha}+\boldsymbol{d}^T\widetilde{\boldsymbol{v}}(\alpha)}\right)$$

so the two vectors

$$\boldsymbol{r}_{\boldsymbol{v}}(\alpha) = \boldsymbol{r}_{\lambda\boldsymbol{x}+(1-\lambda)\boldsymbol{y}}(\alpha) \quad \text{and} \quad \lambda\widetilde{\boldsymbol{x}}(\alpha) + (1-\lambda)\widetilde{\boldsymbol{y}}(\alpha)$$

are parallel to each other (i.e., they are equal up to normalization) because pseudoranks are linear. This simple connection provides a way to compute the strongly preferential PageRank w.r.t. the preference vector $\boldsymbol{v} = \lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}$ just by combining in the same way the pseudoranks of $\boldsymbol{x}$ and $\boldsymbol{y}$, and ℓ_1-normalising the resulting vector. Note that the same process would not work if $\boldsymbol{r}_{\boldsymbol{x}}(\alpha)$ and $\boldsymbol{r}_{\boldsymbol{y}}(\alpha)$ were known *in lieu* of $\widetilde{\boldsymbol{x}}(\alpha)$ and $\widetilde{\boldsymbol{y}}(\alpha)$, as there is no way to recover the (de)normalisation factors.

[11] The reader should note that our formula has some difference in signs w.r.t. the original paper, where it was calculated incorrectly.

7.1 Iterated PageRank

A rather obvious question raising from the view of PageRank as an operator on preference vectors is the behaviour of PageRank w.r.t. *iteration*. What happens if we apply again PageRank to a PageRank vector? We start by approaching the question using weakly preferential PageRank, as the linear dependence on $\boldsymbol{v}$ makes the analysis much easier. To avoid cluttering too much the notation, let us denote with $\boldsymbol{r}^{[k]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ the k-th iteration of weakly preferential PageRank, that is,

$$\begin{aligned}\boldsymbol{r}^{[0]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) &= \boldsymbol{v}\\ \boldsymbol{r}^{[k]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) &= \boldsymbol{r}_{\boldsymbol{r}^{[k-1]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha),\boldsymbol{u}}(\alpha).\end{aligned}$$

Clearly,

$$\boldsymbol{r}^{[k]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = (1-\alpha)^k \boldsymbol{v}(I-\alpha P_{\boldsymbol{u}})^{-k} = (1-\alpha)^k \boldsymbol{v} \sum_{n=0}^{\infty} \binom{n+k-1}{k-1} \alpha^n P^n_{\boldsymbol{u}}.$$

This shows that iterating PageRank is equivalent to choosing a *different damping function* in the sense of [2]. The factor $(I-\alpha P_{\boldsymbol{u}})^{-k}$ strongly resembles the corresponding term in the derivative as obtained in Corollary 3. And indeed, a simple computation shows that for $k > 0$

$$\boldsymbol{r}^{(k)}_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = \frac{k!}{(1-\alpha)^{k+1}} \boldsymbol{r}^{[k+1]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)\big(P^k - P^{k-1}\big), \tag{10}$$

so there is a tight algebraic connection between iteration and derivation. One interesting point is that it might be much quicker to iterate a Gauss–Seidel method and apply the above formula than using the Power Method and the bounds of Theorem 4, at least for small k (albeit upper bounding numerical errors could be difficult).

The same observations hold for pseudoranks: indeed, the above computations are valid also for pseudoranks just by setting $\boldsymbol{u} = \boldsymbol{0}$ (it is easy to check that all results of Section 6 are still valid in this case). However, the situation is completely different for strongly preferential PageRank, where the nonlinear dependency on $\boldsymbol{v}$ makes it difficult to derive similar results: we leave this problem for future work.

There is a final property about equation (10) that we want to highlight. Even if this observation can be stated for the derivatives of any order, let us limit ourselves to first-order derivatives only. Consider the following definition: for every distribution $\boldsymbol{x}$, let us define the *gain vector associated to* $\boldsymbol{x}$ as

$$\Delta\boldsymbol{x} = \boldsymbol{x}(P - I).$$

The gain at each node is the difference between the score that the node would obtain from its in-neighbours and the score that the node actually has; this difference is negative if the node has a score higher than the one its in-neighbours would attribute to it (we might say: if the node is overscored), and positive otherwise (i.e., if the node is underscored).

In the case of first-order derivatives, equation (10) reduces to

$$\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha) = \frac{1}{(1-\alpha)^2} \boldsymbol{r}^{[2]}_{\boldsymbol{v},\boldsymbol{u}}(\alpha)(P - I).$$

That is, the derivative vector $\boldsymbol{r}'_{\boldsymbol{v},\boldsymbol{u}}(\alpha)$ is parallel to $\boldsymbol{r}^{[2]}_{\boldsymbol{v},\boldsymbol{u}}(P - I) = \Delta\boldsymbol{r}^{[2]}_{\boldsymbol{v},\boldsymbol{u}}$. In other words, the first derivative of PageRank at a node is negative iff the node is overscored by the second iterated PageRank; note that there is a shift, here, between the order of differentiation and the number of iterations.

8 Conclusions

We have presented a number of results which outline an analytic study of PageRank with respect to its many parameters. Albeit mainly theoretical in nature, they provide efficient ways to study the global behaviour of PageRank, and dispel a few myths (in particular, about the significance of PageRank when α gets close to 1).

References

[1] Konstantin Avrachenkov, Nelly Litvak, and Kim Son Pham. A singular perturbation approach for choosing PageRank damping factor, 2006.

[2] Ricardo Baeza-Yates, Paolo Boldi, and Carlos Castillo. Generalizing PageRank: damping functions for link-based ranking algorithms. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 308–315, New York, NY, USA, 2006. ACM Press.

[3] Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Ubicrawler: A scalable fully distributed web crawler. *Software: Practice & Experience*, 34(8):711–726, 2004.

[4] Paolo Boldi, Violetta Lonati, Massimo Santini, and Sebastiano Vigna. Graph fibrations, graph isomorphism, and PageRank. *RAIRO Inform. Théor.*, 40:227–253, 2006.

[5] Paolo Boldi, Roberto Posenato, Massimo Santini, and Sebastiano Vigna. Traps and pitfalls of topic-biased PageRank. In *WAW 2006. Fourth Workshop on Algorithms and Models for the Web-Graph*, Lecture Notes in Computer Science. Springer–Verlag, 2007.

[6] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. PageRank as a function of the damping factor. In *Proc. of the Fourteenth International World Wide Web Conference*, pages 557–566, Chiba, Japan, 2005. ACM Press.

[7] Paolo Boldi and Sebastiano Vigna. The WebGraph framework I: Compression techniques. In *Proc. of the Thirteenth International World Wide Web Conference*, pages 595–601, Manhattan, USA, 2004. ACM Press.

[8] Michael Brinkmeier. PageRank revisited. *ACM Transactions on Internet Technology (TOIT)*, 6(3):282–301, 2006.

[9] Gianna Del Corso, Antonio Gullì, and Francesco Romani. Fast PageRank computation via a sparse linear system. *Internet Math.*, 2(3), 2006.

[10] Nadav Eiron, Kevin S. McCurley, and John A. Tomlin. Ranking the web frontier. In *Proceedings of the thirteenth international conference on World–Wide Web*, pages 309–318. ACM Press, 2004.

[11] Gene H. Golub and Chen Greif. Arnoldi-type algorithms for computing stationary distribution vectors, with application to PageRank. Technical Report SCCM-04-15, Stanford University Technical Report, 2004.

[12] Taher Haveliwala and Sepandar Kamvar. The condition number of the PageRank problem. Technical Report 36, Stanford University Technical Report, June 2003.

[13] Taher H. Haveliwala. Efficient computation of PageRank. Technical Report 31, Stanford University Technical Report, October 1999. Available at http://dbpubs.stanford.edu/pub/1999-31.

[14] Taher H. Haveliwala and Sepandar D. Kamvar. The second eigenvalue of the Google matrix. Technical Report 20, Stanford University Technical Report, March 2003. Available at http://dbpubs.stanford.edu/pub/2003-20.

[15] Marius Iosifescu. *Finite Markov Processes and Their Applications*. John Wiley & Sons, 1980.

[16] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 271–279, New York, NY, USA, 2003. ACM Press.

[17] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Exploiting the block structure of the web for computing PageRank. Technical Report 17, Stanford University Technical Report, March 2003. Available at http://dbpubs.stanford.edu/pub/2003-17.

[18] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Extrapolation methods for accelerating PageRank computations. In *Proceedings of the twelfth international conference on World Wide Web*, pages 261–270. ACM Press, 2003.

[19] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tompkins, and Eli Upfal. The Web as a graph. In *PODS '00: Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–10. ACM Press, 2000.

[20] Amy N. Langville and Carl D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):355–400, 2004.

[21] Chris Pan-Chi Lee, Gene H. Golub, and Stefanos A. Zenios. A fast two-stage algorithm for computing PageRank and its extensions. Technical Report SCCM-03-15, Stanford University Technical Report, 2003. Available at http://www-sccm.stanford.edu/pub/sccm/sccm03-15_2.pdf.

[22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 66, Stanford University, 1999. Available at http://dbpubs.stanford.edu/pub/1999-66.

[23] Luca Pretto. A theoretical analysis of google's PageRank. In *Proceedings of the Ninth Symposium on String Processing and Information Retrieval*, pages 131–144, 2002.

[24] Luca Pretto. A theoretical approach to link analysis algorithms, 2002. PhD Thesis.

[25] Stefano Serra-Capizzano. Jordan canonical form of the Google matrix: A potential contribution to the PageRank computation. *SIAM Journal on Matrix Analysis and Applications*, 27(2):305–312, 2005.

[26] Sebastiano Vigna. TruRank: Taking PageRank to the limit. In *Fourteenth International World Wide Web Conference (WWW 2005), Special Interest Tracks & Posters*, pages 976–977, Chiba, Japan, 2005. ACM Press.