

PageRank as a Function of the Damping Factor*

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

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. Recently, however, the behaviour of PageRank with respect to changes in α was discovered to be useful in link-spam detection [21]. Moreover, an analytical justification of the value chosen for α is still missing. In this paper, we give the first 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 an extension of the Power Method that approximates them with convergence $O(r^k \alpha^l)$ for the k -th derivative. Finally, we show a tight connection between iterated computation and analytical behaviour by proving that the k -th iteration of the Power Method gives exactly the PageRank value obtained using a Maclaurin polynomial of degree k . The latter result paves the way towards the application of analytical methods to the study of PageRank.

1 Introduction

PageRank [17] is one of the most important ranking techniques used in today's search engines. Not only is PageRank a simple, robust and reliable way to measure the importance of web pages [3], but it is also computationally advantageous with respect to other ranking techniques in that it is query independent, and content independent. Otherwise said, it can be computed offline using only the web graph¹ structure and then used later, as users submit queries to the search engine, typically aggregated with other, query-dependent rankings [4, 12, 16].

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 on the average. 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, we also allow the surfer to restart with probability $1 - \alpha$ from another node chosen randomly and uniformly, instead of following a link.

As remarked in [5], 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". As the authors of the aforementioned paper, however, we believe that a scientific and detailed study of PageRank is essential to our understanding of the web, and we hope this paper can be a contribution in such program.

*This work has been partially supported by a "Finanziamento per grandi e mega attrezzature scientifiche" of the Università degli Studi di Milano and by the MIUR COFIN Project "Linguaggi formali e automi".

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

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 combining 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 combination 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. When α is 0, the web-graph part of the process is annihilated, resulting in the trivial uniform process. As α goes 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 [19, 18], but also the relative ordering of nodes determined by PageRank can be radically different [14]. The original value suggested by Brin and Page ($\alpha = 0.85$) is the most common choice.

Intuitively, $1 - \alpha$ is an amount of ranking that we agree to give uniformly at each page. This amount will be then funneled through the outlinks of the node. A common form of link spamming funnels carefully this amount towards a single page, giving it 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 it has 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 moreover 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 3 that when α goes to 1 PageRank gets concentrated in the recurrent states, which correspond essentially to the nodes whose strongly connected components have no passage 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-world web graphs the rank of all important nodes (in particular, all nodes of the core component) goes to 0 as α goes to 1.

Thus, PageRank oscillates between a meaningless uniform distribution ($\alpha = 0$) and a meaningless distribution concentrated mostly in irrelevant nodes ($\alpha = 1$). 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. Recently, indeed, a sophisticated form of link-spam detection has been based on the study of the value of PageRank with respect to α [21].

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 , and an iterative algorithm (an extension of the power method) that approximates them.

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* $\alpha \in (0, 1)$. This happens because the k -th coefficient of the Maclaurin series for PageRank can be easily computed during the k -th iteration of the Power Method. This allows to study easily the behaviour of PageRank for any node storing a minimal amount of data.²

²Free Java code implementing all the algorithms described in this paper will be available for download at <http://law.dsi.unimi.it/>.

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). If we want to be specific about the presence of a loop, we shall use the terms *looped* and *loopless*³.

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 link spamming).

If no loopless terminal nodes are present (note that after the preprocessing sketched above they will be the only kind of terminal nodes), we can just normalise uniformly to 1 the row-sums of G by multiplying it by D^{-1} , the inverse of the diagonal degree matrix. However, D is not invertible if loopless terminal nodes are present. The classical way to handle this situation consists in substituting them with nodes that have one outgoing arc toward every node (including the node itself. In other words, in G rows of zeroes are substituted with rows of ones.

Let \bar{G} be the (adjacency matrix of the) resulting graph, and \bar{D} be the diagonal matrix of the outdegrees of \bar{G} (i.e., d_{ii} is the number of ones on the i -th row of \bar{G}). Let also $\mathbf{1}$ be the vector⁴ of all 1's, and \mathbf{v} be any *personalisation vector* (a vector whose elements are all non-negative and sum to 1, which is used to bias PageRank w.r.t. a selected set of trusted pages).

We are providing a toy example in the Appendix that will guide the reader through the paper. In Table 5, the example graph G and its modified version \bar{G} are presented.

In the rest of the paper, we shall use the matrices defined in Figure 1; some of them are functions of the *damping factor* $\alpha \in (0, 1)$, and we will use a notation reflecting this fact. Note that $Q(\alpha)$ is well defined for all $\alpha \in (0, 1)$, as $(I - \alpha P)$ is known to be invertible [20].

$$\begin{aligned} P &= \bar{D}^{-1} \bar{G} \\ A(\alpha) &= \alpha P + (1 - \alpha) \mathbf{1}^T \mathbf{v} \\ C(\alpha) &= I - \alpha P \\ Q(\alpha) &= PC(\alpha)^{-1} \end{aligned}$$

Figure 1: Basic PageRank definitions.

The PageRank vector $\mathbf{r}(\alpha)$ is defined as the dominant eigenvector of $A(\alpha)$; more precisely, as the only vector summing to 1 such that $\mathbf{r}(\alpha)A(\alpha) = \mathbf{r}(\alpha)$. Noting that $\mathbf{r}(\alpha)\mathbf{1}^T = 1$, we get

$$\begin{aligned} \mathbf{r}(\alpha)(\alpha P + (1 - \alpha)\mathbf{1}^T \mathbf{v}) &= \mathbf{r}(\alpha) \\ \alpha \mathbf{r}(\alpha)P + (1 - \alpha)\mathbf{v} &= \mathbf{r}(\alpha) \\ (1 - \alpha)\mathbf{v} &= \mathbf{r}(\alpha)(I - \alpha P), \end{aligned}$$

which yields the following closed formula for PageRank:

$$\mathbf{r}(\alpha) = (1 - \alpha)\mathbf{v}C(\alpha)^{-1}. \quad (1)$$

³In PageRank-related literature, loopless terminal nodes are more commonly known as *dangling nodes*; the same kind of node is often called a *sink* in graph-theoretic literature. Our choice avoids the usage of ambiguous terms that have been given different meanings in different papers.

⁴All vectors in this paper are row vectors.

This is Lemma 3 of [8], albeit in the original statement of this lemma the factor $1 - \alpha$ is missing, probably due to an oversight. Note that (1) can be written as

$$\mathbf{r}(\alpha) = (1 - \alpha)\mathbf{v} \sum_{t=0}^{\infty} (\alpha P)^t,$$

which makes the dependence of PageRank on incoming paths very explicit.

The reader can see the PageRank vector in Figure 7 (the preference vector \mathbf{v} is the uniform vector). PageRank is represented as a function of α in Figure 8.

3 General Behaviour

In this section, we shall discuss the general behaviour of PageRank as a function of the damping factor α , considering in particular what happens when α gets close to 1.

Recall that P (the row-normalised adjacency matrix) is a Markov chain, but in general it is neither aperiodic nor irreducible. Usually, though, in all practical cases P will be aperiodic, but reducible. *In this paper, we shall assume that P is indeed aperiodic.*

Introducing the damping factor has the consequence of obtaining an aperiodic irreducible chain. Indeed, for all $\alpha \in (0, 1)$, $A(\alpha)$ is a Markov chain; moreover, if $\alpha < 1$, $A(\alpha)$ is irreducible and aperiodic. Hence, $A(\alpha)$ admits a *unique* limit distribution $\mathbf{r}(\alpha)$.

3.1 Choosing the damping factor

Clearly, $\mathbf{r}(\alpha)$ is a rational (vector) function of α : usually, though, one looks at $\mathbf{r}(\alpha)$ only for a *specific* value of α . All algorithms to compute PageRank actually compute (or, more precisely, provide an estimate of) $\mathbf{r}(\alpha)$ 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 [17], and it is rumored that Google itself uses this value; it seems that the rankings obtained with this choice are very natural and satisfactory for the users.

Many authors had 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 $A(\alpha)$: indeed, as we remarked in the introduction, $A(0)$ is the uniform matrix and $\mathbf{r}(0)$ is the uniform distribution.

Conversely, as $\alpha \rightarrow 1^-$, the matrix $A(\alpha)$ tends to P : 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 [9] as $\alpha \rightarrow 1^-$, a fact that also influences the other methods used to compute PageRank (which are, after all, variants of the Power Method [17, 7, 6, 15, 11, 10]). 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 [8]).

3.2 Getting close to 1

Even disregarding the problems discussed above, we shall provide convincing reasons that make it inadvisable to use a value of α close to 1.

First observe that, since $\mathbf{r}(\alpha)$ is a rational (coordinatewise) bounded function defined on $(0, 1)$, the limit

$$\mathbf{r}^* = \lim_{\alpha \rightarrow 1^-} \mathbf{r}(\alpha)$$

exists (the reader can see the vector \mathbf{r}^* for our example in the caption of Figure 7).

It is easy to see that \mathbf{r}^* is actually *one of the* limit distributions of P (because $\lim_{\alpha \rightarrow 1^-} A(\alpha) = P$). There are some natural questions about \mathbf{r}^* that we want to address:

- can we somehow characterise the properties of \mathbf{r}^* ?
- what makes \mathbf{r}^* different from the other (infinitely many, if P is reducible) limit distributions of P ?

The first question is the most interesting, because it is about what happens to PageRank when $\alpha \rightarrow 1^-$; 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 limit distribution \mathbf{p} of an aperiodic Markov chain, if $p_x > 0$ then x is recurrent [20].

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 *nonempty* directed path from x to y in G (by nonempty we mean that the path should contain at least one arc).

Clearly, a node is recurrent in P iff $x_{\bar{G}}$ is terminal; otherwise said, x is recurrent (in the Markov chain P) iff $x \rightsquigarrow_{\bar{G}} y$ implies $y \rightsquigarrow_{\bar{G}} x$ as well. Note that nodes with just a loop *are* recurrent (and their component is looped, too).

We now turn to our characterisation theorem, which identifies recurrent states on the basis of G , rather than \bar{G} . The essence of the theorem is that, for what concerns recurrent states, the difference between G and \bar{G} is not significant, unless there are no looped terminal nodes among the components of G . The latter case, however, is as pathological as periodicity in a large web graph.

Theorem 1 *Let G and P be defined as above. Then:*

1. *if G has a component that is looped and terminal (in the component graph), then a node is recurrent for P iff its component is looped and terminal; hence, given any limit distribution \mathbf{p} for P , $p_x > 0$ implies that x is a node of G whose component is looped and terminal;*
2. *if G does not contain a component that is looped and terminal, then every node is recurrent.*

Proof. Note that $x \rightsquigarrow_{\bar{G}} y$ means that there is a nonempty path from x to y in \bar{G} . Such a path can be decomposed into a sequence of (possibly empty) paths in G , from $x = x_0$ to a loopless terminal node y_0 , from a node x_1 to a loopless terminal node y_1 , ..., from a node x_k to $y_k = y$. Moreover, either $k > 0$, or the only path (a path from x to y in G) contains at least one arc.

For case (1), let x be contained in a looped terminal component, and suppose that $x \rightsquigarrow_{\bar{G}} y$. By the observation above, this path in \bar{G} can be decomposed into a sequence of paths of G towards loopless terminal nodes, plus a final path to y : but from x you cannot reach a loopless terminal node of G (because x is contained in a looped terminal component), so the path is simply a nonempty path of G , i.e., $x \rightsquigarrow_G y$. But then y is in the same component as x , so $y \rightsquigarrow_G x$ as well, and we obtain the result. For the converse, suppose that x is not in a looped terminal component: we will show that there is a y such that $x \rightsquigarrow_{\bar{G}} y$ but not $y \rightsquigarrow_{\bar{G}} x$. We distinguish two cases:

- suppose that there is a looped terminal component that can be reached from x_G in the component graph of G ; let y be any node in such component. Clearly $x \rightsquigarrow_G y$, and hence $x \rightsquigarrow_{\bar{G}} y$, but $y \rightsquigarrow_{\bar{G}} x$ does not hold (from y you can only reach nodes of y_G both in G and in \bar{G});
- otherwise, suppose that there is a loopless terminal y such that $x \rightsquigarrow_G y$ (or $x = y$ if x itself is terminal); let z be any node in a looped terminal component G : now $x \rightsquigarrow_{\bar{G}} z$ (you first go from x to y and then you “jump” to z), but from z you cannot reach x (because x is not in the same component).

For case (2), take any two nodes x and y of G . In the component graph of G there will be two terminal components x'_G and y'_G that are reachable from x_G and y_G , respectively. Both are, by hypothesis, loopless. In other words, there are two terminal nodes x' and y' such that $x \rightsquigarrow_G x'$ (or $x = x'$) and $y \rightsquigarrow_G y'$ (or $y = y'$). This means that $x \rightsquigarrow_{\bar{G}} y$ and vice versa, unless $x = y$ (and both are terminal), in which case again both $x \rightsquigarrow_{\bar{G}} y$ and vice versa. ■

The statement of the previous theorem may seem a bit unfathomable. The essence, however, could be stated as follows: except for strongly connected graphs, or graphs all whose terminal components are trivial and loopless, the recurrent nodes are exactly those whose component is looped and terminal. These nodes are often called *rank sinks*, as they absorb all the rank circulating through the graph.

As we remarked, a real-world graph will certainly contain at least one looped terminal component, so the first statement of the theorem will hold. This means that *most* nodes x will be such that $r_x^* = 0$. In particular, this will be true of all the nodes in the core component [13]: 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 example).

This is a rather convincing justification that, contradicting the common beliefs, choosing α too close to 1 *does not* provide any good PageRank. Rather, PageRank becomes “sensible” somewhere in between 0 and 1.

As far as the second question is concerned, we provide a

Conjecture 1 \mathbf{r}^* is the limit distribution of P when the starting distribution is uniform, that is,

$$\lim_{\alpha \rightarrow 1^-} \mathbf{r}(\alpha) = \lim_{n \rightarrow \infty} \frac{\mathbf{1}}{N} P^n.$$

Note that the conjecture is trivial when P is irreducible, because in that case P has but one stationary distribution.

4 Derivatives

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

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

4.1 Exact formulae

The main objective of this section is providing exact formulae for the derivatives of $\mathbf{r}(-)$. Define $\mathbf{r}'(\alpha)$, $\mathbf{r}''(\alpha)$, \dots , $\mathbf{r}^{(k)}(\alpha)$ as the first, second, \dots , k -th derivative of $\mathbf{r}(\alpha)$ with respect to α .

We start by providing the basic relations between these vector functions:

Theorem 2 *The following identities hold:*

1. $\mathbf{r}'(\alpha) = (\mathbf{r}(\alpha)P - \mathbf{v})C(\alpha)^{-1}$;
2. for all $k > 0$, $\mathbf{r}^{(k+1)}(\alpha) = (k+1)\mathbf{r}^{(k)}(\alpha)PC(\alpha)^{-1}$.

Proof. Multiplying (1) by $C(\alpha)$ and differentiating memberwise:

$$\mathbf{r}(\alpha)C'(\alpha) + \mathbf{r}'(\alpha)C(\alpha) = -\mathbf{v} \quad (2)$$

$$\mathbf{r}'(\alpha)C(\alpha) = -\mathbf{r}(\alpha)C'(\alpha) - \mathbf{v} \quad (3)$$

$$\mathbf{r}'(\alpha)C(\alpha) = \mathbf{r}(\alpha)P - \mathbf{v}. \quad (4)$$

Since $C(\alpha)$ is invertible:

$$\mathbf{r}'(\alpha) = (\mathbf{r}(\alpha)P - \mathbf{v})C(\alpha)^{-1}.$$

Moreover, differentiating once more (4), we obtain:

$$\begin{aligned} \mathbf{r}'(\alpha)C'(\alpha) + \mathbf{r}''(\alpha)C(\alpha) &= \mathbf{r}'(\alpha)P \\ \mathbf{r}''(\alpha)C(\alpha) &= \mathbf{r}'(\alpha)P - \mathbf{r}'(\alpha)C'(\alpha) \\ \mathbf{r}''(\alpha)C(\alpha) &= \mathbf{r}'(\alpha)P + \mathbf{r}'(\alpha)P \end{aligned}$$

hence

$$\mathbf{r}''(\alpha) = 2\mathbf{r}'(\alpha)PC(\alpha)^{-1},$$

which accounts for the base case ($k = 1$) of an induction for the second statement. For the inductive step, again multiplying by $C(\alpha)$ and differentiating memberwise:

$$\begin{aligned} \mathbf{r}^{(k+2)}(\alpha)C(\alpha) + \mathbf{r}^{(k+1)}(\alpha)C'(\alpha) &= (k+1)\mathbf{r}^{(k+1)}(\alpha)P \\ \mathbf{r}^{(k+2)}(\alpha)C(\alpha) &= \mathbf{r}^{(k+1)}(\alpha)[(k+1)P - C'(\alpha)] \end{aligned}$$

and the thesis follows easily. ■

We can reformulate the statement concerning the first-order derivative as follows:

Corollary 1 *The following identity holds:*

$$\mathbf{r}'(\alpha) = \mathbf{r}(\alpha) \left(Q(\alpha) - \frac{1}{1-\alpha} I \right).$$

Proof. From Theorem 2, we obtain $\mathbf{r}'(\alpha) = \mathbf{r}(\alpha)PC(\alpha)^{-1} - \mathbf{v}C(\alpha)^{-1}$. Using (1) we can rewrite this as $\mathbf{r}(\alpha)PC(\alpha)^{-1} - \frac{1}{1-\alpha}\mathbf{r}(\alpha)$, hence the result. ■

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

Corollary 2 For every $k > 0$

$$\mathbf{r}^{(k)}(\alpha) = k (\mathbf{r}(\alpha)P - \mathbf{v})C(\alpha)^{-1}Q(\alpha)^{k-1}$$

or, equivalently,

$$\mathbf{r}^{(k)}(\alpha) = k \mathbf{r}(\alpha) \left(Q(\alpha) - \frac{1}{1-\alpha}I \right) Q(\alpha)^{k-1}.$$

Proof. Just proceed from Theorem 2 by iterate substitution, and finally apply Corollary 1. ■

4.2 Approximating the derivatives

The formulae obtained in Section 4.1 do not lead directly to an effective algorithm that computes derivatives: even assuming that the exact value of $\mathbf{r}(\alpha)$ is available, to obtain the derivatives one should invert $C(\alpha)$ (see Theorem 2), 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 for a given value of α , and we will show how these approximations converge to the desired vectors.

The simplest and most important algorithm that computes PageRank [17] is an application of the Power Method; the algorithm computes a sequence of vectors $\mathbf{v}_0, \mathbf{v}_1, \dots$ where $\mathbf{v}_0 = \mathbf{v}$ and $\mathbf{v}_{k+1} = \mathbf{v}_k A(\alpha)$. This sequence of vectors converges to $\mathbf{r}(\alpha)$, and convergence speed depends on α ; more precisely, the difference in norm between the k -th iterate and the exact value is $O(\alpha^k)$. In practice, the algorithm provides good approximation quickly: in the original paper [17] the authors state that 40 to 50 iterations are enough on reasonable data sets; of course, more sophisticated approaches have been proposed in the literature to reduce the number of iterations and/or the amount of computation needed at each iteration [17, 7, 6, 15, 11, 10], but they are basically all variants of the Power Method.

The reader can see the first few iterates of the Power Method applied to our example in Figure 1.

We are going to present a modified version of the basic algorithm that will compute PageRank and its derivatives up to (any desired) index K , and to do this it will use $K + 1$ vectors. In other words, it will build $K + 1$ vector sequences: the sequence $\mathbf{s}_0^{(0)}(\alpha), \mathbf{s}_1^{(0)}(\alpha), \dots, \mathbf{s}_t^{(0)}(\alpha), \dots$ will be used to approximate $\mathbf{r}(\alpha)$ (and will be defined exactly as in the classical PageRank algorithm); the sequence $\mathbf{s}_0^{(1)}(\alpha), \mathbf{s}_1^{(1)}(\alpha), \dots, \mathbf{s}_t^{(1)}(\alpha), \dots$ will be used to approximate $\mathbf{r}'(\alpha)$; and so on. Note that the sequence $\mathbf{s}_0^{(k)}(\alpha), \mathbf{s}_1^{(k)}(\alpha), \dots, \mathbf{s}_t^{(k)}(\alpha), \dots$ will not, in general, converge to $\mathbf{r}^{(k)}(\alpha)$ *per se*; rather, there will be an associated sequence $\mathbf{q}_0^{(k)}(\alpha), \mathbf{q}_1^{(k)}(\alpha), \dots, \mathbf{q}_t^{(k)}(\alpha), \dots$ based on it, that will actually converge to the desired derivative.

The vector sequences are defined in Figure 2. Note that only the $K + 1$ vectors $\mathbf{s}_t^{(k)}(\alpha)$ ($0 \leq k \leq K$) need to be stored, whereas $\mathbf{q}_t^{(k)}(\alpha)$ are only defined for convenience, and can be implemented, for example, as a function.

Our first result is about convergence of the first-order derivative.⁵

Theorem 3 $\lim_{t \rightarrow \infty} \mathbf{q}_t^{(1)}(\alpha) = \mathbf{r}'(\alpha)$, and the difference in norm is $O(t\alpha^t)$, that is:

$$\left\| \mathbf{q}_t^{(1)}(\alpha) - \mathbf{r}'(\alpha) \right\| = O(t\alpha^t) \quad \text{as } t \rightarrow \infty.$$

⁵We do not assume a particular norm—all our proofs are correct with any p -norm. We just note that $\|S\|_1 = 1$ for any stochastic S (as we use row vectors), and so $\|P^t\|_p$ is bounded by a constant that depends on P 's size, but not on P 's elements or on t .

$$\begin{aligned}
\mathbf{s}_0^{(0)}(\alpha) &= \mathbf{v} \\
\mathbf{s}_{t+1}^{(0)}(\alpha) &= \mathbf{s}_t^{(0)}(\alpha)A(\alpha) \\
\mathbf{s}_0^{(k+1)}(\alpha) &= \mathbf{q}_0^{(k)}(\alpha) \\
\mathbf{s}_{t+1}^{(k+1)}(\alpha) &= \alpha\mathbf{s}_t^{(k+1)}(\alpha)P + \mathbf{q}_t^{(k)}(\alpha)P \\
\\
\mathbf{q}_t^{(0)}(\alpha) &= \mathbf{s}_t^{(0)}(\alpha) \\
\mathbf{q}_t^{(1)}(\alpha) &= \mathbf{s}_t^{(1)}(\alpha) - \frac{1}{1-\alpha}\mathbf{s}_t^{(0)}(\alpha) \\
\mathbf{q}_t^{(k)}(\alpha) &= k\mathbf{s}_t^{(k)}(\alpha) \quad \text{for all } k \geq 2.
\end{aligned}$$

Figure 2: Basic definitions for the approximation algorithm.

Proof. Recall that $\|\mathbf{q}_t^{(0)}(\alpha) - \mathbf{r}(\alpha)\| = O(\alpha^t)$, since the second eigenvalue of $A(\alpha)$ is at most α [9]. An easy proof by induction shows that for all t

$$\mathbf{s}_t^{(1)}(\alpha) = \mathbf{v}(\alpha P)^t + \sum_{s=0}^{t-1} \mathbf{q}_{t-s-1}^{(0)}(\alpha)(\alpha P)^s P,$$

so

$$\mathbf{q}_t^{(1)}(\alpha) = \mathbf{v}(\alpha P)^t + \sum_{s=0}^{t-1} \mathbf{q}_{t-s-1}^{(0)}(\alpha)(\alpha P)^s P - \frac{1}{1-\alpha}\mathbf{q}_t^{(0)}(\alpha).$$

Now, by Corollary 1, we have:

$$\mathbf{r}'(\alpha) = \mathbf{r}(\alpha) \left(Q(\alpha) - \frac{1}{1-\alpha}I \right).$$

Since

$$Q(\alpha) = PC(\alpha)^{-1} = P(I - \alpha P)^{-1} = \sum_{s=0}^{\infty} (\alpha P)^s P,$$

we have

$$\mathbf{r}'(\alpha) = \sum_{s=0}^{\infty} \mathbf{r}(\alpha)(\alpha P)^s P - \frac{1}{1-\alpha}\mathbf{r}(\alpha).$$

Hence, we can bound the convergence rate as follows:

$$\begin{aligned}
\|\mathbf{r}'(\alpha) - \mathbf{q}_t^{(1)}(\alpha)\| &\leq \left\| \sum_{s=t}^{\infty} \mathbf{r}(\alpha)(\alpha P)^s P \right\| + \\
&\quad + \left\| \sum_{s=0}^{t-1} \left(\mathbf{r}(\alpha) - \mathbf{q}_{t-s-1}^{(0)}(\alpha) \right) (\alpha P)^s P \right\| + \\
&\quad \frac{1}{1-\alpha} \|\mathbf{r}(\alpha) - \mathbf{q}_t^{(0)}(\alpha)\| + \|\mathbf{v}(\alpha P)^t\|.
\end{aligned}$$

We provide an upper bound for each of the summands above. As far as the first summand is concerned, recall that $\sum_{t=n}^{\infty} X^t = (I - X)^{-1} X^n$ (whenever the first series converges). Thus,

$$\left\| \sum_{s=t}^{\infty} \mathbf{r}(\alpha)(\alpha P)^s P \right\| \leq \left\| (I - \alpha P)^{-1} \right\| \cdot \left\| (\alpha P)^t \right\| \cdot \left\| \mathbf{r}(\alpha) P \right\|,$$

and the first summand is $O(\alpha^t)$. The second summand can be bounded as follows:

$$\left\| \sum_{s=0}^{t-1} \left(\mathbf{r}(\alpha) - \mathbf{q}_{t-s-1}^{(0)}(\alpha) \right) (\alpha P)^s P \right\| = O(\alpha^t) \left\| \sum_{s=0}^{t-1} P^{s+1} \right\| = O(\alpha^t) \sum_{s=0}^{t-1} O(1) = O(t\alpha^t).$$

The third and the fourth summands are both $O(\alpha^t)$. All summands are thus $O(t\alpha^t)$ —hence the result. ■

As far as the other derivatives are concerned, we have:

Theorem 4 For every $k > 1$, $\lim_{t \rightarrow \infty} \mathbf{q}_t^{(k)}(\alpha) = \mathbf{r}^{(k)}(\alpha)$, and the difference in norm is $O(t^k \alpha^t)$.

Proof. First of all, by induction on k and t , one can prove that

$$\mathbf{s}_t^{(k)}(\alpha) = (k-1) \frac{\alpha}{\alpha-1} \mathbf{v}(\alpha P)^t + \sum_{s=0}^{t-1} \mathbf{q}_{t-s-1}^{(k-1)}(\alpha) (\alpha P)^s P,$$

for all $k > 1$ and all t . The base case $t = 0$ is trivial (by an easy induction on all $k > 1$), whereas the case $k = 2$ can be obtained by induction on t using Theorem 3, noting that the rule for computing $\mathbf{q}_t^{(1)}(\alpha)$ is a special case. The inductive step is then obtained using the rule that defines $\mathbf{s}_{t+1}^{(k+1)}(\alpha)$.

Now, recalling that (from Theorem 2)

$$\mathbf{r}^{(k)}(\alpha) = k \mathbf{r}^{(k-1)}(\alpha) P C(\alpha)^{-1} = k \sum_{s=0}^{\infty} \mathbf{r}^{(k-1)}(\alpha) (\alpha P)^s P,$$

we have

$$\begin{aligned} \left\| \mathbf{r}^{(k)}(\alpha) - \mathbf{q}_t^{(k)}(\alpha) \right\| &= \left\| k \sum_{s=0}^{\infty} \mathbf{r}^{(k-1)}(\alpha) (\alpha P)^s P - k \mathbf{s}_t^{(k)}(\alpha) \right\| \leq \\ &k \left\| \sum_{s=t}^{\infty} \mathbf{r}^{(k-1)}(\alpha) (\alpha P)^s P \right\| + k \frac{\alpha}{1-\alpha} \left\| \mathbf{v}(\alpha P)^t \right\| + \\ &+ k \left\| \sum_{s=0}^{t-1} \left(\mathbf{r}^{(k-1)}(\alpha) - \mathbf{q}_{t-s-1}^{(k-1)}(\alpha) \right) (\alpha P)^s P \right\|. \end{aligned}$$

The result follows along the lines of the last part of the proof of Theorem 3. ■

We remark two important points that deserve further analysis. First of all, the big-oh notation hides a number of constants independent of t . However, when k is large or α very close to 1 these constants may become important. Second, we did not give a detailed evaluation of the numerical precision that is necessary to perform these computations.

4.3 Implementation of the algorithm

The results of the previous section can be used to modify the classical PageRank algorithm, based on the Power Method, so to compute an approximation of the derivatives of PageRank up to a certain index.

The algorithm uses a vector $s-, -$ where the first index represents the derivative index (from 0 to K , inclusive, where K is the highest derivative order to be computed) and the second index represents the node. In other words, at step t the vector $sk, -$ represents $s_t^{(k)}(\alpha)$. The vector $q_t^{(k)}(\alpha)$ is not itself represented as a vector, but rather it is implemented by the procedure $q()$.

The procedure $init()$ initialises the vector $s-, -$, whereas $step()$ computes the vector for the next iteration (the new vector is indicated by $s'-, -$).

The stopping criterion can be decided in many ways: for example, at each step, the norms of the differences between each derivative and the derivative at the previous step are computed, and the iteration is stopped if all such norms are below a certain threshold.

```
procedure q(k, i)
  if k=0 then return s0, i;
  else if k=1 then return s1, i - s0, i/(1 -  $\alpha$ );
  else return k · sk, i;

procedure init()
  for i:=0, 1, ..., N - 1 do
    s0, i:=vi;
    for k:=1, ..., K do
      sk, i:=q(k - 1, i);
    end for
  end for

procedure step()
  for i:=0, 1, ..., N - 1 do
    for k:=0, 1, ..., K do
      s'k, i = 0;
    end for
  end for
  for i:=0, 1, ..., N - 1 do
    d:=outdegree of node i;
    for all successors j of i do
      s'0, j:=s'0, j +  $\alpha$  + q(0, i)/d;
      for k:= 1, 2, ..., K do
        s'k, j:=s'k, j + ( $\alpha$ sk, i + q(k - 1, i))/d;
      end for
    end for
  end for

procedure computePageRankAndDerivatives()
  init();
  do step(); while not stopping condition;
```

5 Maclaurin series

The rational function $r(-)$ can be expressed using its Maclaurin series (e.g., the Taylor series about 0); let us denote by $t_n(\alpha)$ the n -th degree Maclaurin polynomial of $r(-)$ evaluated in α .

Clearly, Maclaurin polynomials offer an appealing way to study PageRank in relation to α . To obtain an explicit formula for $\mathbf{t}_n(\alpha)$, just recall from Corollary 2 that $\mathbf{r}^{(k)}(0) = k \mathbf{r}(0)(Q(0) - I)Q(0)^{k-1}$. Since $\mathbf{r}(0) = \mathbf{v}$ and $Q(0) = P$, we have, for all $k > 0$,

$$\mathbf{r}^{(k)}(0) = k \mathbf{v}(P - I)P^{k-1}.$$

Now, since $\mathbf{t}_n(\alpha) = \sum_{k=0}^n (1/k)\alpha^k \mathbf{r}^{(k)}(0)$, we have

$$\mathbf{t}_n(\alpha) = \mathbf{v} \left(I + \sum_{k=1}^n \alpha^k (P^k - P^{k-1}) \right).$$

Two important problems face us now: first of all, how to compute $\mathbf{t}_n(\alpha)$; second, how to choose n . Both problems will be solved by a surprisingly simple relationship between Maclaurin polynomials and the Power Method that will be proved in this section. To obtain our main result, we will need the following:

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

$$(\lambda M + (1 - \lambda)R)^n = \lambda^n M^n + (1 - \lambda) \sum_{k=0}^{n-1} \lambda^k R M^k.$$

or, equivalently,

$$(\lambda M + (1 - \lambda)R)^n = \lambda^n M^n + R(I - \lambda^n M^{n-1}) + R \sum_{k=1}^{n-1} \lambda^k (M^k - M^{k-1}).$$

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

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

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

Of course, the last result can be easily restated in any \mathbf{R} -algebra.

We can now come to the main result of this section, which equates analytic approximation (the degree of the Maclaurin polynomial) with computational approximation (the number of iterations of the Power Method):

Theorem 5 *The n -th approximation of PageRank computed by the Power Method with damping factor α coincides with the n -th degree Maclaurin polynomial of PageRank evaluated in α . In other words, $\mathbf{v}A(\alpha)^n = \mathbf{t}_n(\alpha)$.*

Proof. Apply Lemma 1 to the case when $M = P$, $R = \mathbf{1}^T \mathbf{v}$ and $\lambda = \alpha$. We have:

$$A(\alpha)^n = \alpha^n P^n + \mathbf{1}^T \mathbf{v} - \alpha^n \mathbf{1}^T \mathbf{v} P^{n-1} + \mathbf{1}^T \mathbf{v} \sum_{k=1}^{n-1} \alpha^k (P^k - P^{k-1}),$$

hence

$$\begin{aligned} \mathbf{q}_n^{(0)}(\alpha) = \mathbf{v}A(\alpha)^n &= \alpha^n \mathbf{v}P^n + \mathbf{v} - \alpha^n \mathbf{v}P^{n-1} + \mathbf{v} \sum_{k=1}^{n-1} \alpha^k (P^k - P^{k-1}) = \\ &= \mathbf{v} + \mathbf{v} \sum_{k=1}^{n-1} \alpha^k (P^k - P^{k-1}) = \mathbf{t}_n(\alpha). \end{aligned}$$

■

As a consequence:

Corollary 3 *The difference between the k -th and the $(k - 1)$ -th approximation of PageRank (as computed by the Power Method), divided by α^k , is the k -th coefficient⁶ of the Maclaurin series of PageRank.*

The previous corollary is apparently innocuous. However, as a consequence the data obtained computing PageRank for a given α can be used to compute immediately PageRank for *any other* α , obtaining the result of the Power Method after the same number of iterations. Indeed, by saving the Maclaurin coefficients 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 (lowering of course by k the approximating polynomial). Note, however, that the algorithm presented in Section 4.3 provides values of the derivatives for a specific α *with a precision guarantee*.

The first few coefficients of the Maclaurin polynomial for our example are shown in Figure 2.

6 Experimental results

Figure 3 illustrates from an experimental viewpoint the convergence speed theorems of Section 4.2. We computed PageRank and its derivatives up to index four (inclusive) and we plotted the difference, in L_2 -norm (we used for the computation a small, 325 557-nodes graph of the sites of the Italian CNR), between two successive iterates during the first 70 iterations; for every derivative we also show the upper bounds proved in Theorems 3 and 4. Note that there is a transient irregular behaviour due to the constants hidden in the proofs.

Figure 4 shows the convergence of Maclaurin polynomials toward the actual PageRank behaviour for a chosen node. Finally, in Figure 9 we display the approximation obtained with a 100-degree Maclaurin polynomial. 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 [1] and indexed by WebGraph [2].

⁶The coefficients are vectors, because we are approximating a vector function.

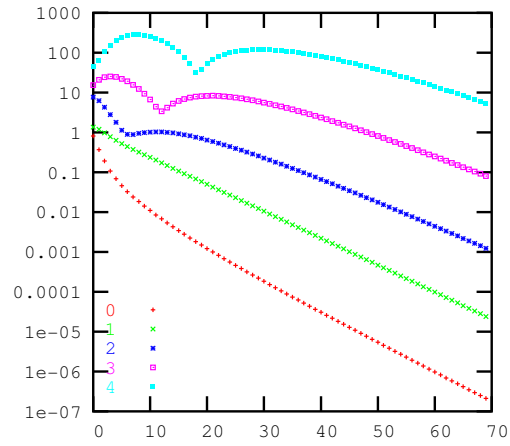


Figure 3: The convergence speed in the computation of derivatives up to order 4 (the label is the order of the derivative).

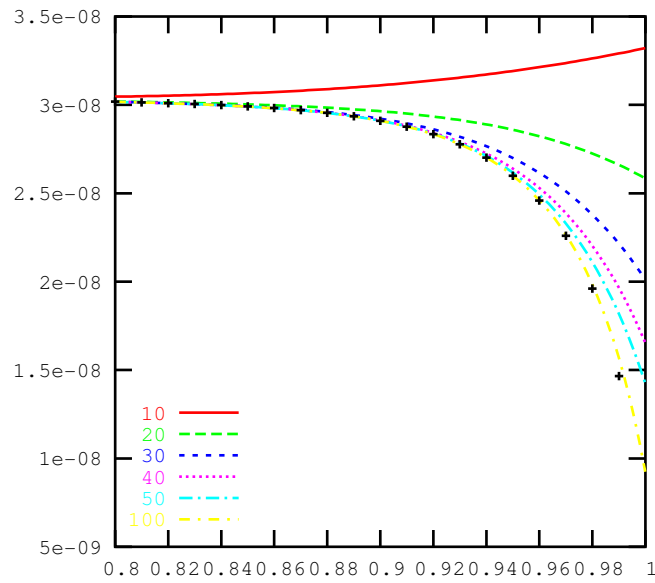


Figure 4: Approximating $r(\alpha)$ for a specific node (cross-shaped points) using Maclaurin polynomials of different degrees (shown in the legend).

7 Conclusions

We have presented a number of results which outline the first analytic study of PageRank when the damping factor changes. While our results are 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).

A last point that is worth being noted is that our algorithm to obtain the Maclaurin polynomials for PageRank may be used to determine new forms of ranking; for example, one may define the *total rank* of a page x as $\int_0^1 r_x(\alpha) d\alpha$. This quantity (the area under the PageRank curve of node x) is independent from α , and induces interesting rankings that will be studied in a forthcoming paper.

References

- [1] Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Ubcrawler: A scalable fully distributed web crawler. *Software: Practice & Experience*, 34(8):711–726, 2004.
- [2] 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.
- [3] Soumen Chakrabarti, Byron Dom, David Gibson, Jon Kleinberg, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Hypersearching the web. *Scientific American*, June 1999.
- [4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [5] 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.
- [6] 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.
- [7] Taher Haveliwala. Efficient computation of PageRank. Technical report, Stanford University Technical Report, October 1999.
- [8] Taher Haveliwala and Sepandar Kamvar. The condition number of the PageRank problem. Technical Report 36, Stanford University Technical Report, June 2003.
- [9] Taher Haveliwala and Sepandar Kamvar. The second eigenvalue of the Google matrix. Technical Report 20, Stanford University Technical Report, March 2003.
- [10] Sepandar Kamvar, Taher Haveliwala, Christopher Manning, and Gene Golub. Exploiting the block structure of the web for computing PageRank. Technical Report 17, Stanford University Technical Report, March 2003.
- [11] 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.

- [12] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999.
- [13] 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.
- [14] Amy N. Langville and Carl D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):355–400, 2004.
- [15] Chris Pan-Chi Lee, Gene H. Golub, and Stefanos A. Zenios. A fast two-stage algorithm for computing PageRank and its extensions. Technical report, Stanford University Technical Report, 2004.
- [16] Ronny Lempel and Shlomo Moran. SALSA: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, 19(2):131–160, 2001.
- [17] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, Stanford University, Stanford, CA, USA, 1998.
- [18] 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.
- [19] Luca Pretto. A theoretical approach to link analysis algorithms, 2002. PhD Thesis.
- [20] Eugene Seneta. *Non-Negative Matrices and Markov Chains*. Springer Series in Statistics. Springer-Verlag, 1981.
- [21] Hui Zhang, Ashish Goel, Ramesh Govindan, Kahn Mason, and Benjamin Van Roy. Making eigenvector-based reputation systems robust to collusion. In Stefano Leonardi, editor, *Proceedings WAW 2004*, number 3243 in LNCS, pages 92–104. Springer-Verlag, 2004.

Appendix: An example

To clarify the discussion of the previous section, we provide a full example in Figure 5. Node 4 is the only terminal node of the graph, but nodes 4 and 5 belong to a looped terminal component (see Figure 6). Correspondingly, Figure 8 shows that PageRank for nodes 4 and 5 grows, whereas for all other nodes goes to 0 as $\alpha \rightarrow 1^-$. Note, however, the maximum attained by node 0 at $\alpha \approx 0.7$.

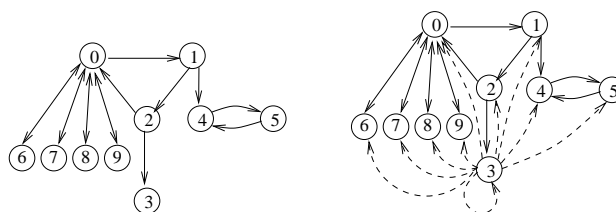


Figure 5: A graph with $n = 10$ nodes and its modified version.

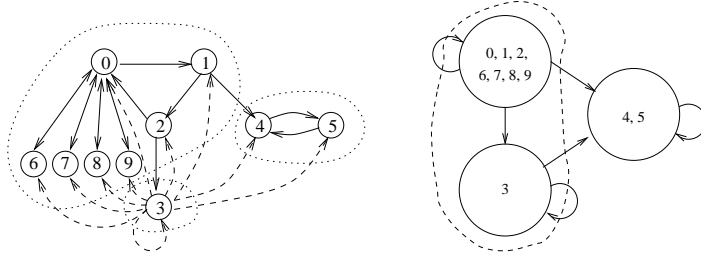


Figure 6: The components of the graph in Figure 5 and the corresponding component graph. The dashed line in the component graph gathers components that are merged in \bar{G} . In both G and \bar{G} the only terminal component is $\{4, 5\}$.

Step	Approximation
0	$\langle 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100, 0.100 \rangle$
1	$\langle 0.415, 0.049, 0.075, 0.075, 0.113, 0.070, 0.049, 0.049, 0.049, 0.049 \rangle$
2	$\langle 0.232, 0.100, 0.051, 0.062, 0.078, 0.072, 0.100, 0.100, 0.100, 0.100 \rangle$
3	$\langle 0.391, 0.066, 0.070, 0.049, 0.097, 0.056, 0.066, 0.066, 0.066, 0.066 \rangle$
4	$\langle 0.283, 0.093, 0.054, 0.056, 0.076, 0.063, 0.093, 0.093, 0.093, 0.093 \rangle$

Table 1: The approximations computed in the first iterations of the Power Method (with $\alpha = 0.85$).

$$\mathbf{r}(\alpha) = \left\langle -5 \frac{(-1+\alpha)(\alpha^2+18\alpha+4)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, -2 \frac{(-1+\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, 2 \frac{(-1+\alpha)(7\alpha^2-5\alpha-10)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, \right. \\ \left. \frac{(-1+\alpha)(11\alpha^2+8\alpha^3-10\alpha-20)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, -\frac{\alpha^4+16\alpha^3+14\alpha^2-30\alpha-20}{(\alpha+1)(8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200)}, -\frac{15\alpha^3+6\alpha^2-20\alpha-20}{(\alpha+1)(8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200)}, \right. \\ \left. -2 \frac{(-1+\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, -2 \frac{(-1+\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, -2 \frac{(-1+\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200}, -2 \frac{(-1+\alpha)(10+2\alpha+\alpha^2)}{8\alpha^4+\alpha^3-170\alpha^2-20\alpha+200} \right\rangle$$

Figure 7: The explicit formula for PageRank as a function of α . Its limit is $\lim_{\alpha \rightarrow 1^-} \mathbf{r}(\alpha) = \mathbf{r}^* = \langle 0, 0, 0, 0, 1/2, 1/2, 0, 0, 0, 0 \rangle$.

	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 2: The coefficients of the Maclaurin series.

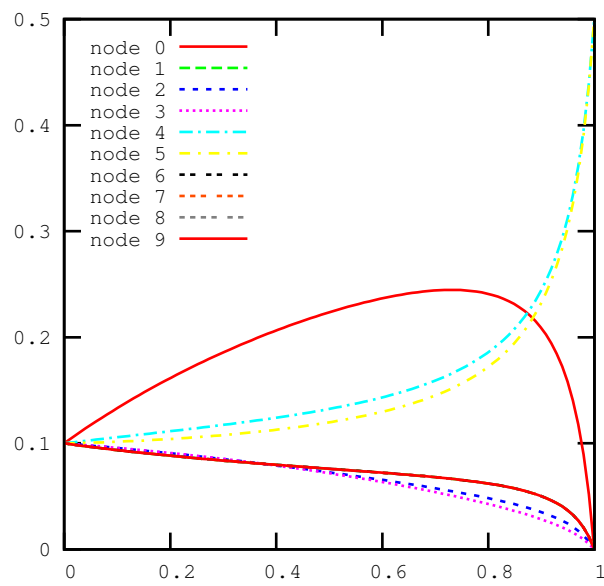


Figure 8: The behaviour of the components of $\mathbf{r}(\alpha)$. They all go to zero except for nodes 4 and 5—the only nodes belonging to a terminal component. Note, however, the maximum attained by node 0 at $\alpha \approx 0.7$.

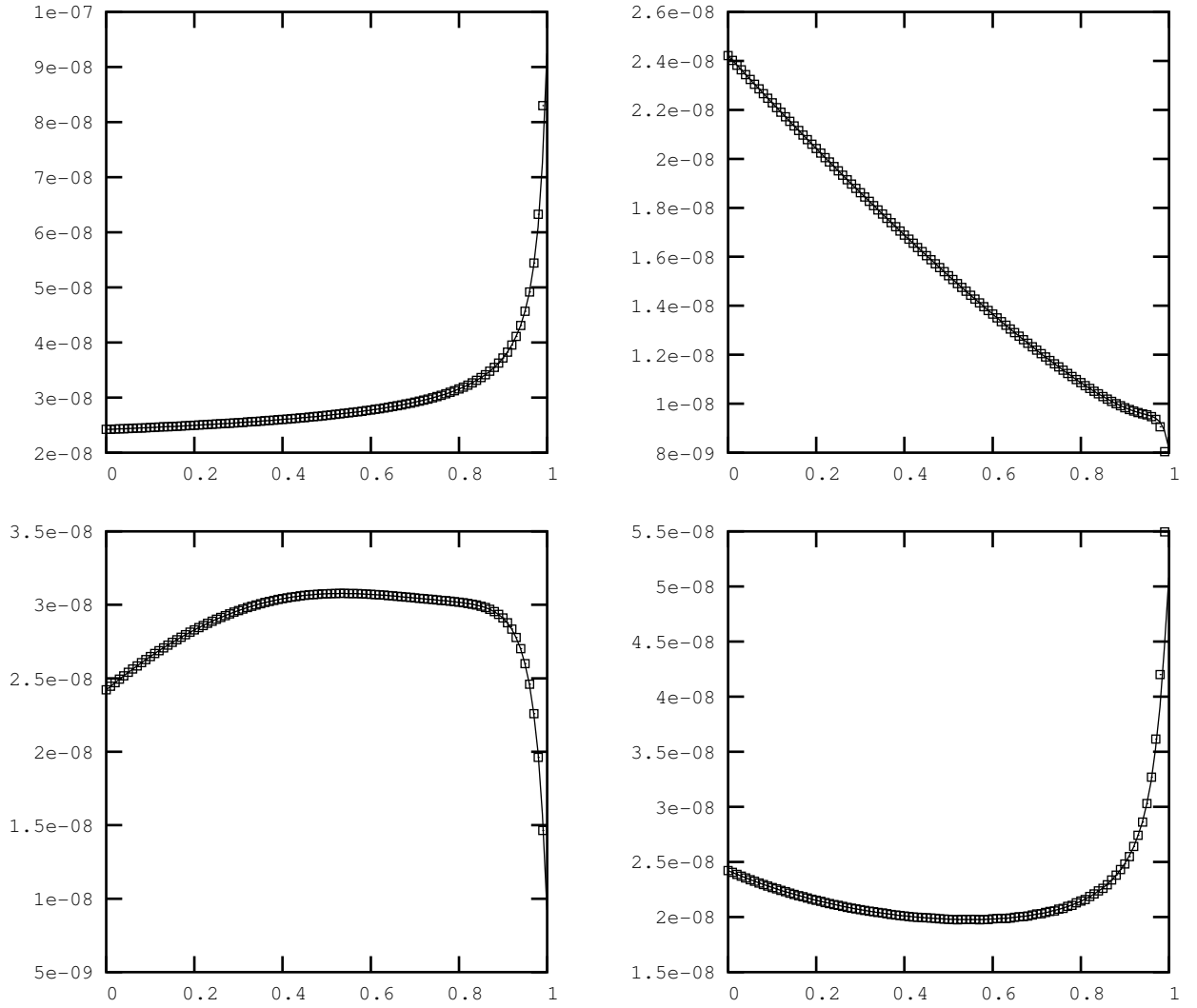


Figure 9: Examples of approximations obtained using a Maclaurin polynomial of degree 100, for nodes with different behaviours (the points were tabulated by computing PageRank explicitly with 100 regularly spaced values of α).