PageRank in scale-free random graphs*

Ningyuan Chen[1], Nelly Litvak[2], Mariana Olvera-Cravioto[1]

[1] Columbia University, 500 W. 120th Street, 3rd floor, New York, NY 10027
[2] University of Twente, P.O.Box 217, 7500AE, Enschede, The Netherlands

Abstract. We analyze the distribution of PageRank on a directed configuration model and show that as the size of the graph grows to infinity it can be closely approximated by the PageRank of the root node of an appropriately constructed tree. This tree approximation is in turn related to the solution of a linear stochastic fixed point equation that has been thoroughly studied in the recent literature.

1 Introduction

Google's PageRank proposed by Brin and Page [5] is arguably the most influential technique for computing centrality scores of nodes in a network. Numerous applications include graph clustering [3], spam detection [9], and citation analysis [8, 20]. In this paper we analyze the power law behavior of PageRank scores in scale-free directed random graphs.

In real-world networks, it is often found that the fraction of nodes with (in- or out-) degree k is $\approx c_0 k^{-\alpha-1}$, usually $\alpha \in (1,3)$, see e.g. [17] for an excellent review of the mathematical properties of complex networks. More than ten years ago Pandurangan et al. [16] discovered the interesting fact that PageRank scores also exhibit power laws, with the same exponent as the in-degree. This property holds for a broad class of real-life networks [19]. In fact, the hypothesis that this always holds in power-law networks is plausible.

However, analytical mathematical evidence supporting this hypothesis is surprisingly scarce. As one of the few examples, Avrachenkov and Lebedev [4] obtained the power law behavior of average PageRank scores in a preferential attachment graph by using Polya's urn scheme and advanced numerical methods.

In a series of papers, Volkovich et al. [14, 19, 18] suggested an analytical explanation for the power law behavior of PageRank by comparing it to the endogenous solution of a stochastic fixed point equation (SFPE). The properties of this equation and the study of its multiple solutions has itself been an interesting topic in the recent literature [2, 10, 11, 12, 15, 1], and is related to the broader study of weighted branching processes. The tail behavior of the endogenous solution, the one more closely related to PageRank, was given in [10, 11, 12, 15], where it was shown to have a power law under many different sets of assumptions. However, the SFPE does not fully explain the behavior of PageRank in

* This research is partially funded by the EU-FET Open grant NADINE (288956).

networks since it implicitly assumes that the underlying graph is an infinite tree, an assumption that is not in general satisfied in real-world networks.

This paper makes a fundamental step further by extending the analysis of PageRank to graphs that are not necessarily trees. Specifically, we assume that the underlying graph is a directed configuration model (DCM) with given degree distributions, as developed by Chen and Olvera-Cravioto [7]. We present numerical evidence that in this type of graphs the behavior of PageRank is very close to the one on trees. Intuitively, this is true for two main reasons: 1) the influence of remote nodes on the PageRank of an arbitrary node decreases exponentially fast with the graph distance; and 2) the DCM is asymptotically tree-like, that is, when we explore a graph starting from a given node, then with high probability the first loop is observed at a distance of order $\log n$, where n is the size of the graph (see Figure 1). Our main result establishes analytically that PageRank in a DCM is well approximated by the PageRank of the root node of a suitably constructed tree as the graph size goes to infinity.

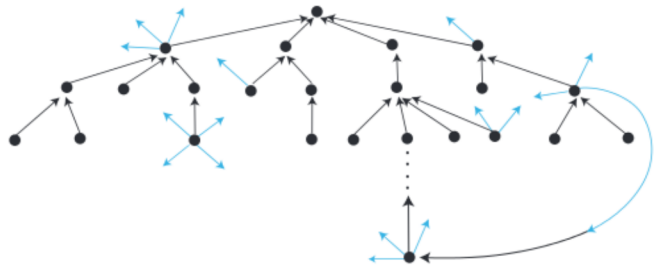

Fig. 1. Graph construction process. Unpaired outbound stubs are in blue.

Section 2 below describes the DCM as presented in [7]. Then, in Section 3 we analytically compare the PageRank scores in the DCM to their approximate value obtained after a finite number of power iterations. Next, in Section 4 we explain how to couple the PageRank of a randomly chosen node with the root node of a suitable branching tree, and give our main analytical results. Finally, in Section 5 we give numerical results validating our analytical work. The complete proofs for more general stochastic recursions, that also cover the PageRank case considered here, will be given in our upcoming paper [6].

2 Directed Random Graphs

We will give below an algorithm, taken from [7], that can be used to generate a scale-free directed graph. Formally, power law distributions are modeled using the mathematical notion of regular variation. A nonnegative random variable X is said to be regularly varying, if $\overline{F}(x) := P(X > x) = L(x)x^{-\alpha}$, $x > 0$, where $L(\cdot)$ is a slowly varying function, that is, $\lim_{x\to\infty} L(tx)/L(x) = 1$, for all $t > 0$.

Our goal now is to create a directed graph $\mathcal{G}(n)$ with the property that the in-degrees and out-degrees will be approximately distributed, for large sizes of

the graph, according to distributions $f_k^{\text{in}} = P(\mathsf{N} = k)$, and $f_k^{\text{out}} = P(\mathsf{D} = k)$, $k = 0, 1, 2, 3, \ldots$, respectively, where $E[\mathsf{N}] = E[\mathsf{D}]$. The only condition needed is that these distributions satisfy

$$\overline{F^{\text{in}}}(x) = \sum_{k>x} f_k^{\text{in}} \le x^{-\alpha} L_{\text{in}}(x) \qquad \text{and} \qquad \overline{F^{\text{out}}}(x) = \sum_{k>x} f_k^{\text{out}} \le x^{-\beta} L_{\text{out}}(x),$$

for some slowly varying functions $L_{\text{in}}(\cdot)$ and $L_{\text{out}}(\cdot)$, and $\alpha, \beta > 1$.

The first step in our procedure is to generate an appropriate bi-degree sequence

$$(\mathsf{N}_n, \mathsf{D}_n) = \{(N_i, D_i) : 1 \le i \le n\}$$

representing the n nodes in the graph. The algorithm given below will ensure that the in- and out-degrees follow closely the desired distributions and also that the sums of in- and out-degrees are the same:

$$L_n := \sum_{i=1}^{n} N_i = \sum_{i=1}^{n} D_i.$$

Denote

$$\kappa_0 = \min\{1 - \alpha^{-1}, 1 - \beta^{-1}, 1/2\}.$$

Algorithm 1. *Generation of a bi-degree sequence with given in-/out-degree distributions.*

1. Fix $0 < \delta_0 < \kappa_0$.
2. Sample an i.i.d. sequence $\{\mathsf{N}_1, \ldots, \mathsf{N}_n\}$ from distribution F^{in}.
3. Sample an i.i.d. sequence $\{\mathsf{D}_1, \ldots, \mathsf{D}_n\}$ from distribution F^{out}, independent of $\{\mathsf{N}_i\}$.
4. Define $\Delta_n = \sum_{i=1}^{n} (\mathsf{N}_n - \mathsf{D}_n)$. If $|\Delta_n| \le n^{1-\kappa_0+\delta_0}$ proceed to step 5; otherwise repeat from step 2.
5. Choose randomly $|\Delta_n|$ nodes $\{i_1, i_2, \ldots, i_{|\Delta_n|}\}$ without replacement and let

$$N_i = \begin{cases} \mathsf{N}_i + 1 & \text{if } \Delta_n < 0 \text{ and } i \in \{i_1, i_2, \ldots, i_{\Delta_n}\}, \\ \mathsf{N}_i & \text{otherwise,} \end{cases}$$

$$D_i = \begin{cases} \mathsf{D}_i + 1 & \text{if } \Delta_n \ge 0 \text{ and } i \in \{i_1, i_2, \ldots, i_{\Delta_n}\}, \\ \mathsf{D}_i & \text{otherwise.} \end{cases}$$

Remark: It was shown in [7] that

$$P\left(|\Delta_n| > n^{1-\kappa_0+\delta_0}\right) = O\left(n^{-\delta_0(\kappa_0-\delta_0)/(1-\kappa_0)}\right) \tag{1}$$

as $n \to \infty$, and therefore the Algorithm 1 will always terminate after a finite number of steps (i.e., it will eventually proceed to step 5).

Having obtained a realization of the bi-degree sequence $(\mathsf{N}_n, \mathsf{D}_n)$, we now use the configuration model to construct the random graph. The idea in the directed case is essentially the same as for undirected graphs. To each node v_i

we assign N_i inbound half-edges and D_i outbound half-edges; then, proceed to match inbound half-edges to outbound half-edges to form directed edges. To be more precise, for each unpaired inbound half-edge of node v_i choose randomly from all the available unpaired outbound half-edges, and if the selected outbound half-edge belongs to node, say, v_j, then add a directed edge from v_j to v_i to the graph; proceed in this way until all unpaired inbound half-edges are matched. Note that the resulting graph is not necessarily simple, i.e., it may contain self-loops and multiple edges in the same direction.

We point out that conditional on the graph being simple, it is uniformly chosen among all simple directed graphs having bi-degree sequence $(\mathsf{N}_n, \mathsf{D}_n)$ (see [7]). Moreover, it was also shown in [7] that, provided $\alpha, \beta > 2$, the probability of obtaining a simple graph through this procedure is bounded away from zero, and therefore one can obtain a simple graph having $(\mathsf{N}_n, \mathsf{D}_n)$ as its bi-degree sequence by simply repeating the algorithm enough times. When we can only ensure that $\alpha, \beta > 1$, then a simple graph can still be obtained without loosing the distributional properties of the in- and out-degrees by erasing the self-loops and merging multiple edges in the same direction. These considerations about the graph being simple are nonetheless irrelevant to the current paper.

3 PageRank iterations in the DCM

Although PageRank can be thought of as the solution to a system of linear equations, we will show in this section how it is sufficient to consider only a finite number of power iterations to obtain an accurate approximation for the PageRank of all the nodes in the graph. We first introduce some notation.

Let $M = M(n) \in \mathsf{R}^{n \times n}$ be matrix constructed as follows:

$$M_{i,j} = \begin{cases} s_{ij}c/D_i, & \text{if there are } s_{ij} \text{ edges from } i \text{ to } j, \\ 0, & \text{otherwise,} \end{cases}$$

and let $\mathbf{1}$ be the row vector of ones. In the classical definition [13], PageRank $\pi = (\pi_1, \ldots, \pi_n)$ is the unique solution to the following equation:

$$\pi = \pi(cM) + \frac{1-c}{n}\mathbf{1}, \tag{2}$$

where $c \in (0,1)$ is a constant known as the damping factor. Rather than analyzing π directly, we consider instead its scale-free version

$$n\pi =: \mathsf{R} = \mathsf{R}\,(cM) + (1-c)\mathbf{1} \tag{3}$$

obtained by multiplying (2) by the size of the graph n. Moreover, whereas π_i is a probability distribution ($\pi_i \geq 0$ for all i and $\pi \mathbf{1}^T = 1$), its scale-free version $\mathsf{R} = (R_1, \ldots, R_n)$ has components that are essentially unbounded for large n and that satisfy $E[R_i] = 1$ for all n and all $1 \leq i \leq n$ (hence the name scale-free).

One way to solve the system of linear equations given in (3) is via power iterations. We define the kth iteration of PageRank on the graph as follows.

First initialize PageRank with a vector $\mathsf{r}_0 = r_0\mathsf{1}$, $r_0 \geq 0$, and then iterate according to $\mathsf{R}^{(n,0)} = \mathsf{r}_0$ and

$$\mathsf{R}^{(n,k)} = \mathsf{R}^{(n,k-1)}M + (1-c)\mathsf{1} = (1-c)\mathsf{1}\sum_{i=0}^{k-1} M^i + \mathsf{r}_0 M^k$$

for $k \geq 1$. In this notation, $\mathsf{R} = \mathsf{R}^{(n,\infty)}$, and our main interest is to analyze the distribution of the PageRank of a randomly chosen node in the DCM, say $R_1^{(n,\infty)}$. The first step of the analysis is to compare $\mathsf{R}^{(n,\infty)}$ to its kth iteration $\mathsf{R}^{(n,k)}$. To this end, note that $\mathsf{R}^{(n,\infty)} = (1-c)\mathsf{1}\sum_{i=0}^{\infty} M^i$, and therefore,

$$\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)} = \mathsf{r}_0 M^k - (1-c)\mathsf{1}\sum_{i=k}^{\infty} M^i.$$

Moreover,

$$\begin{aligned}\left\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}\right\|_1 &\leq \left\|\mathsf{r}_0 M^k\right\|_1 + (1-c)\sum_{i=0}^{\infty}\left\|\mathsf{1}M^{k+i}\right\|_1 \\ &\leq r_0 n \left\|M^k\right\|_\infty + (1-c)n\sum_{i=0}^{\infty}\left\|M^{k+i}\right\|_\infty,\end{aligned}$$

where for the last inequality we used the observation that

$$\left\|\mathsf{1}M^r\right\|_1 = \sum_{j=1}^{n}\sum_{i=1}^{n}(M^r)_{ij} = \sum_{i=1}^{n}\left\|(M^r)_{i\bullet}\right\|_1 \leq n\left\|M^r\right\|_\infty,$$

where $A_{i\bullet}$ denotes the ith row of matrix A. Furthermore, since M is equal to c times an adjacency matrix, we have

$$\left\|M^r\right\|_\infty \leq \|M\|_\infty^r = c^r.$$

It follows that

$$\left\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}\right\|_1 \leq r_0 n c^k + (1-c)n\sum_{i=0}^{\infty} c^{k+i} = (r_0+1)nc^k. \tag{4}$$

In general networks, the inequality for the L_1-norm (4) does not provide information on convergence of specific coordinates and does not give a good upper bound for the quantity $|R_1^{(n,k)} - R_1^{(n,\infty)}|$ that we are interested in. However, the DCM has the additional property that all coordinates of the vector $\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}$ have the same distribution, since by construction, the DCM makes all permutations of the nodes' labels equally likely. This leads to the following observation.

Let $\mathcal{F}_n = \sigma((\mathsf{N}_n, \mathsf{D}_n))$ denote the sigma-algebra generated by the bi-degree sequence, which does not include information about the pairing process. Then, conditional on $\mathcal{F}_n$,

$$E\left[\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right| \middle| \mathcal{F}_n\right] = \frac{1}{n}E\left[\left\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,\infty)}\right\|_1 \middle| \mathcal{F}_n\right] \leq (r_0+1)\,c^k,$$

and Markov's inequality gives,

$$P\left(\left|R_1^{(n,\infty)} - R_1^{(n,k)}\right| > \epsilon\right) \leq E\left[\epsilon^{-1} E\left[\left|R_1^{(n,k)} - R_1^{(n,\infty)}\right| \middle| \mathcal{F}_n\right]\right]$$
$$\leq (r_0 + 1)\,\epsilon^{-1} c^k \tag{5}$$

for any $\epsilon > 0$.

Note that (5) is a probabilistic statement, which is not completely analogous to (4). In fact, (5) states that we can achieve any level of precision with a pre-specified high probability by simply increasing the number of iterations k. This leads to the following heuristic, that if the DCM looks locally like a tree for k generations, where k is the number of iterations needed to achieve the desired precision in (5), then the PageRank of node 1 in the DCM will be essentially the same as the PageRank of the root node of a suitably constructed tree. The precise result and a sketch of the arguments will be given in the next section.

4 Main Result: Coupling with a thorny branching tree

As mentioned in the previous section, we will now show how to identify $R_1^{(n,k)}$ with the PageRank of the root node of a tree. To start, we construct a variation of a branching tree where each node has an edge pointing to its parent but also has number of outbound stubs or half-edges that are pointing outside of the tree (i.e., to some auxiliary node). We will refer to this tree as a Thorny Branching Tree (TBT), the name "thorny" referring to the outbound stubs (see Figure 1).

To construct simultaneously the graph $\mathcal{G}(n)$ and the TBT, denoted by $\mathcal{T}$, we start by choosing a node uniformly at random, and call it node 1 (the root node). This first node will have N_1 inbound stubs which we will proceed to match with randomly chosen outbound stubs. These outbound stubs are sampled independently and with replacement from all the possible $L_n = \sum_{i=1}^{n} D_i$ outbound stubs, discarding any outbound stub that has already been matched. This corresponds to drawing independently at random from the distribution

$$\begin{aligned} f_n(i,j) &= P(\text{node has } i \text{ offspring, } j \text{ outbound links } |\mathcal{F}_n) \\ &= \sum_{k=1}^{n} 1(N_k = i, D_k = j) P(\text{an outbound stub of node } k \text{ is sampled } |\mathcal{F}_n) \\ &= \sum_{k=1}^{n} 1(N_k = i, D_k = j) \frac{D_k}{L_n}. \end{aligned} \tag{6}$$

This is a so-called size-biased distribution, since nodes with more outbound stubs are more likely to be chosen.

To keep track of which outbound stubs have already been matched we will label them 1, 2, or 3 according to the following rule:

1. Outbound stubs with label 1 are stubs belonging to a node that is not yet attached to the graph.

2. Outbound stubs with label 2 belong to nodes that are already part of the graph but that have not yet been paired with an inbound stub.
3. Outbound stubs with label 3 are those which have already been paired with an inbound stub and now form an edge in the graph.

Let Z_r, $r \geq 0$, denote the number of inbound stubs of all the nodes in the graph at distance r of the first node. Note that $Z_0 = N_1$ and Z_r is also the number of nodes at distance $(r+1)$ of the first node.

To draw the graph we initialize the process by labeling all outbound stubs with a 1, except for the D_1 outbound stubs of node 1 that receive a 2. We then start by pairing the first of the N_1 inbound stubs with a randomly chosen outbound stub, say belonging to node j. Then node j is attached to the graph by forming an edge with node 1, and all the outbound stubs from the new node are now labeled 2. In case that $j = 1$ the pairing forms a self-loop and no new nodes are added to the graph. Next, we label the chosen outbound stub with a 3, since it has already been paired, and in case $j \neq 1$, give all the other outbound stubs of node j a label 2. We continue in this way until all N_1 inbound stubs of node 1 have been paired, after which we will be left with Z_1 unmatched inbound stubs that will determine the nodes at distance 2 from node 1. In general, the kth iteration of this process is completed when all Z_{k-1} inbound stubs have been matched with an outbound stub, and the process ends when all L_n inbound stubs have been paired. Note that whenever an outbound stub with label 2 is chosen a cycle or double edge is formed in the graph. If at any point we sample an outbound stub with label 3 we simply discard it and do a redraw until we obtain an outbound stub with labels 1 or 2.

We now explain the coupling with the TBT. We start with the root node (node 1, generation 0) that has $\hat{N}_1 = N_1$ offspring. Let $\hat{Z}_k$ denote the number of individuals in generation $k+1$ of the tree, $\hat{Z}_0 = \hat{N}_1$. For $k \geq 1$, each of the $\hat{Z}_{k-1}$ individuals in the kth generation will independently have offspring and outbound stubs according to the random joint distribution $f_n(i,j)$ given in (6).

The coupling of the graph and the TBT is done according to the following rules:

1. If an outbound stub with label 1 is chosen, then both the graph and the TBT will connect the chosen outbound stub to the inbound stub being matched, resulting in a node being added to the graph and an offspring being born to its parent. In particular, if the chosen outbound stub corresponds to node j, then the new offspring in the TBT will have $D_j - 1$ outbound stubs (pointing to the auxiliary node) and N_j inbound stubs (number of offspring). We then update the labels by giving a 2 label to all the 'sibling' outbound stubs of the chosen outbound stub, and a 3 label to the chosen outbound stub itself.
2. If an outbound stub with label 2 is sampled it means that its corresponding node already belongs to the graph, and a cycle, self-loop, or multiple edge is created. In $\mathcal{T}$, we proceed as if the outbound stub had label 1 and create a new node, which is a copy of the drawn node. The coupling between DCM and TBT breaks at this point.

3. If an outbound stub with label 3 is drawn it means that this stub has already been matched, and the coupling breaks as well. In $\mathcal{T}$, we again proceed as if the outbound stub had had a label 1. In the graph we do a redraw.

Note that the processes Z_k and $\hat{Z}_k$ are identical as long as the coupling holds. Showing that the coupling holds for a sufficient number of generations is the essence of our main result.

Definition 1 Let τ be the number of generations in the TBT that can be completed before the first outbound stub with label 2 or 3 is drawn, i.e., $\tau = k$ if the first inbound stub to draw an outbound stub with label 2 or 3 belonged to a node i, such that the graph distance between i and the root node is exactly k.

The following result gives us an estimate as to when the coupling between the exploration process of the graph and the construction of the tree is expected to break.

Lemma 1. *Suppose $(\mathsf{N}_n, \mathsf{D}_n)$ are constructed using Algorithm 1 with $\alpha > 1$, and $\beta > 2$. Let $\mu = E[\mathsf{N}] = E[\mathsf{D}] > 1$. Then, for any $1 \le k \le h \log n$ with $0 < h < 1/(2\log\mu)$ there exists a $\delta > 0$ such that,*

$$P(\tau \le k) = O\left(n^{-\delta}\right) \qquad \text{as } n \to \infty.$$

The proof of Lemma 1 is rather technical, so we will only provide a sketch in this paper. The detailed proof will be given in [6].

Proof (Qualitative argument). Let $\hat{V}_s$ be the number of outbound stubs of all nodes in generation s of the tree. The intuition behind the proof is that for all $s = 1, 2, \ldots$, neither $\hat{Z}_s$, nor $\hat{V}_s$ are expected to be much larger than their means:

$$E\left[\hat{Z}_s \middle| \mathcal{F}_n\right] \approx \mu^{s+1} \qquad \text{and} \qquad E\left[\hat{V}_s \middle| \mathcal{F}_n\right] \approx \lambda\mu^s,$$

where $\lambda = E[\mathsf{D}^2]/\mu$. Next, note that an inbound stub of a node in the rth generation will be the first one to be paired with an outbound stub having label 2 or 3 with a probability bounded from above by

$$P_r := \frac{1}{L_n} \sum_{s=0}^{r} \hat{V}_s \approx \frac{\lambda\mu^r}{n(\mu-1)}.$$

Furthermore, for event $\{\tau = r\}$ to occur one of the $\hat{Z}_r$ inbound stubs must have been paired with an outbound stub with labels 2 or 3, which is bounded by the probability that a Binomial random variable with parameters $(\hat{Z}_r, P_r)$ is greater or equal than 1. Since $P_r = o(1)$ for $r \le k$, this probability is $1 - (1-P_r)^{\hat{Z}_r} = P_r\hat{Z}_r(1 + O(P_r)) = O\left(\mu^{2r}n^{-1}\right)$.

Formally, to ensure that the approximations given above are valid, we first show that the event

$$E_k = \left(\max_{0\le r\le k} \mu^{-r}\hat{Z}_r\mu^r \le x_n, \max_{0\le r\le k} \mu^{-(r+1)} \sum_{s=0}^{r} \hat{V}_s \le x_n \right)$$

occurs with high probability as $n \to \infty$ for a suitably chosen $x_n \to \infty$. Then, sum over $r = 0, 1, \ldots, k$ the events $\{\tau = r, E_k\}$ to obtain that $P(\tau \leq k, E_k) = O\left(\mu^{2k} n^{-1}\right)$, which goes to zero for $k \leq h \log n$.

Our main result is now a direct consequence of the bound derived in (5) and Lemma 1 above, since before the coupling breaks $R_1^{(n,k)}$ and the PageRank, computed after k iterations, of the root node of the coupled tree coincide.

Theorem 1. *Suppose $(\mathsf{N}_n, \mathsf{D}_n)$ are constructed using Algorithm 1 with $\alpha > 1$, and $\beta > 2$. Let $\mu = E[\mathsf{N}\,] = E[\mathsf{D}] > 1$ and $c \in (0,1)$. Then, for any $\epsilon > 0$ and any $1 \leq k \leq h \log n$ with $0 < h < 1/(2 \log \mu)$ there exists a $\delta > 0$ such that,*

$$P\left(\left|R_1^{(n,\infty)} - \hat{R}_1^{(n,k)}\right| > \epsilon\right) \leq (r_0 + 1)\epsilon^{-1} c^k + O\left(n^{-\delta}\right),$$

as $n \to \infty$, where $\hat{R}_1^{(n,k)}$ is the PageRank, after k iterations, of the root node of the TBT described above.

In the forthcoming paper [6] we explore further the distribution of the PageRank of the root node of $\mathcal{T}$ and show that $\hat{R}_1^{(n,k)}$ converges to the endogenous solution of a SFPE on a weighted branching tree, as originally suggested in [14, 19, 18]. Moreover, the tail behavior of this solution has been fully described in [18, 10, 11].

5 Numerical Results

In this last section we give some numerical results showing the accuracy of the TBT approximation to the PageRank in the DCM. To generate the bi-degree sequence we use as target distributions two Pareto-like distributions. More precisely, we set

$$\mathsf{N}_i = \lfloor X_{1,i} + Y_{1,i} \rfloor, \quad \mathsf{D}_i = \lfloor X_{2,i} + Y_{2,i} \rfloor,$$

where the $\{X_{1,i}\}$ and the $\{X_{2,i}\}$ are independent sequences of i.i.d. Pareto random variables with shape parameters $\alpha > 1$ and $\beta > 2$, respectively, and scale parameters $x_1 = (\alpha - 1)/\alpha$ and $x_2 = (\beta - 1)/\beta$, respectively (note that $E[X_{1,i}] = E[X_{2,i}] = 1$ for all i). The sequences $\{Y_{1,i}\}$ and $\{Y_{2,i}\}$ are independent sequences, each consisting of i.i.d. exponential random variables with means $1/\lambda_1 > 0$ and $1/\lambda_2$, respectively. The addition of the exponential random variables allows more flexibility in the modeling of the in- and out-degree distributions while preserving a power law tail behavior; the parameters λ_1, λ_2 are also used to match the means $E[\mathsf{N}\,]$ and $E[\mathsf{D}]$.

Once the sequences $\{\mathsf{N}_i\}$ and $\{\mathsf{D}_i\}$ are generated, we use Algorithm 1 to obtain a valid bi-degree sequence $(\mathsf{N}_n, \mathsf{D}_n)$. Given this bi-degree sequence we next proceed to construct the graph and the TBT simultaneously, according to the rules described in Section 4. To compute $\mathsf{R}^{(n,\infty)}$ we perform power iterations with $r_0 = 1$ until $\|\mathsf{R}^{(n,k)} - \mathsf{R}^{(n,k-1)}\|_2 < \epsilon_0$ for some tolerance ϵ_0. We only

generate the TBT for the required number of generations in each of the examples; the computation of $\hat{R}_1^{(n,k)}$ can be done recursively starting from the leaves using

$$\hat{R}_i^{(n,0)} = 1, \quad \hat{R}_i^{(n,k)} = \sum_{j \to i} c\hat{R}_j^{(n,k-1)} + (1-c), \quad k > 0, \tag{7}$$

where $j \to i$ means that node j is an offspring of node i.

Tables 1-3 below compare the PageRank of node 1 in the graph, $R_1^{(n,\infty)}$, the PageRank of node 1 only after k power iterations, $R_1^{(n,k)}$, and the PageRank of the root node of the coupled tree after the same k generations, $\hat{R}_1^{(n,k)}$. The magnitude of the mean squared errors (MSEs), computed using $R_1^{(n,\infty)}$ as the true value, is also given in each table. The tolerance for computing $R_1^{(n,\infty)}$ is set to $\varepsilon_0 = 10^{-6}$. For each n, we generate 100 realizations of $\mathcal{G}(n)$ as well as of the corresponding TBTs and take the empirical average of the PageRank values and of the MSEs. Table 1 includes results for different sizes of the graph, and uses $k_n = \lfloor \log n \rfloor$ iterations for the finite approximations. We note that all the MSEs clearly decrease as n increases since k_n also increases with n.

n	$R_1^{(n,1)}$	$R_1^{(n,k_n)}$	$\hat{R}_1^{(n,k_n)}$	MSE for $R_1^{(n,k_n)}$	MSE for $\hat{R}_1^{(n,k_n)}$
10	0.931	0.946	0.983	3.90E-03	4.20E-02
100	1.023	1.027	1.068	1.80E-04	3.70E-02
1000	1.000	1.002	1.010	1.20E-05	8.00E-04
10000	0.964	0.965	0.962	1.00E-06	7.50E-04

Table 1. $\delta = 2$, $\bullet = 2{:}5$, $\blacklozenge_1 = 1$, $c = 0{:}5$, $k_n = \lfloor \log n \rfloor$.

Table 2 illustrates the impact of using different values of k, with the error between $R_1^{(n,k)}$ and $R_1^{(n,\infty)}$ clearly decreasing as k increases. The simulations were run on a graph with $n = 10,000$ nodes. We also point out that although the accuracy of finitely many PageRank iterations improves as k gets larger, the MSE of the tree approximation seems to plateau after a certain point. In order to obtain a higher level of precision we also need to increase the size of the graph (as suggested by Theorem 1).

k_n	$R_1^{(n,1)}$	$R_1^{(n,k_n)}$	$\hat{R}_1^{(n,k_n)}$	MSE for $R_1^{(n,k_n)}$	MSE for $\hat{R}_1^{(n,k_n)}$
2	0.908	0.933	0.928	7.1E-03	8.59E-03
4	0.929	0.933	0.933	1.5E-04	2.20E-04
6	0.908	0.909	0.910	5.4E-06	5.08E-05
8	0.883	0.884	0.884	8.8E-08	1.20E-06
10	0.948	0.949	0.950	7.6E-09	8.16E-05
15	0.932	0.932	0.932	7.9E-13	2.89E-05

Table 2. $n = 10000$, $\delta = 2$, $\bullet = 2{:}5$, $\blacklozenge_1 = 1$, $c = 0{:}5$.

Table 3 shows the same comparison as in Table 2, for fixed n, for different values of the damping factor c. As c gets larger, the approximations provided by both $R_1^{(n,k_n)}$ and $\hat{R}_1^{(n,k_n)}$ get worse due to the slower convergence of PageRank.

c	$R_1^{(n,1)}$	$R_1^{(n,k_n)}$	$\hat{R}_1^{(n,k_n)}$	MSE for $R_1^{(n,k_n)}$	MSE for $\hat{R}_1^{(n,k_n)}$
0.1	1.011	1.011	1.011	3.8E-22	3.33E-09
0.3	0.958	0.958	0.958	9.8E-13	1.91E-07
0.5	0.898	0.898	0.899	2.7E-08	2.63E-06
0.7	0.755	0.757	0.760	2.4E-05	2.03E-04
0.9	0.663	0.764	0.799	8.3E-02	1.25E-01

Table 3. n = 10000, = 2, = 2:5, = 1, k_n = blog nc = 9.

Our last numerical result shows how the distribution of PageRank on the TBT approximates the distribution of PageRank on the DCM. To illustrate this we generated a graph with $n = 100$ nodes and parameters $\alpha = 2$, $\beta = 2.5$, $\mu = 3$ and $c = 0.5$. We set the number of PageRank iterations (number of generations in the TBT) to be $k = 4$. We then computed the empirical CDFs of the PageRank of all nodes in the graph and that of the PageRank after only k iterations. We also generated the coupled TBT 1000 times based on the same graph; each time by randomly choosing some node i to be the root and computing $\hat{R}_i^{(n,k)}$ according to (7). Figure 2 plots the empirical CDF of PagerRank on $\mathcal{G}(n)$, the empirical CDF of PageRank on $\mathcal{G}(n)$ after only k iterations, and the empirical CDF of the PageRank of the 1000 root nodes after the same k iterations. We can see that the CDFs of PageRank on $\mathcal{G}(n)$ after a finite number of iterations and that of the true PageRank on $\mathcal{G}(n)$ are almost indistinguishable. The PageRank on the TBT also approximates this distribution quite well, especially considering that $n = 100$ is not particularly large.

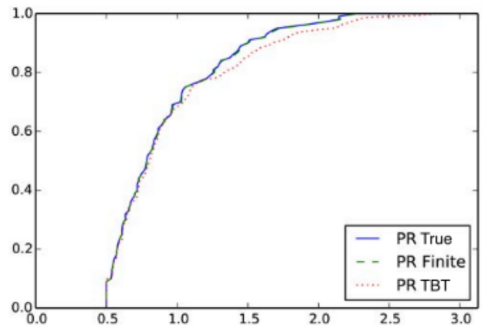

Fig. 2. The empirical distributions of PageRank on G(n) (true and after finitely many iterations) and the empirical distribution of the PageRank of the root in the TBT.

Bibliography

[1] G. Alsmeyer, E. Damek, and S. Mentemeier. Tails of fixed points of the two-sided smoothing transform. In Springer Proceedings in Mathematics & Statistics: Random Matrices and Iterated Random Functions, 2012.

[2] G. Alsmeyer and M. Meiners. Fixed points of the smoothing transform: Two-sided solutions. Probab. Theory Relat. Fields, 155(1-2):165–199, 2013.

[3] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In Proceedings of FOCS2006, pages 475–486, 2006.

[4] K. Avrachenkov and D. Lebedev. PageRank of scale-free growing networks. Internet Mathematics, 3(2):207–231, 2006.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 33:107–117, 1998.

[6] N. Chen, N. Litvak, and M. Olvera-Cravioto. Ranking algorithms on directed configuration networks. Technical report, 2014.

[7] N. Chen and M. Olvera-Cravioto. Directed random graphs with given degree distributions. Stochastic Systems, 3(1):147–186, 2013.

[8] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding scientific gems with Googles PageRank algorithm. Journal of Informetrics, 1(1):8–15, 2007.

[9] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web spam with TrustRank. In Proceeding of VLDB2004, pages 576–587, 2004.

[10] P.R. Jelenković and M. Olvera-Cravioto. Information ranking and power laws on trees. Adv. Appl. Prob., 42(4):1057–1093, 2010.

[11] P.R. Jelenković and M. Olvera-Cravioto. Implicit renewal theory and power tails on trees. Adv. Appl. Prob., 44(2):528–561, 2012.

[12] P.R. Jelenković and M. Olvera-Cravioto. Implicit renewal theory for trees with general weights. Stochastic Process. Appl., 122(9):3209–3238, 2012.

[13] A.N. Langville and C.D. Meyer. Google PageRank and beyond. Princeton University Press, 2006.

[14] N. Litvak, W.R.W. Scheinhardt, and Y. Volkovich. In-degree and PageRank: Why do they follow similar power laws? Internet mathematics, 4(2):175–198, 2007.

[15] M. Olvera-Cravioto. Tail behavior of solutions of linear recursions on trees. Stochastic Process. Appl., 122(4):1777–1807, 2012.

[16] G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to characterize Web structure. Internet Mathematics, 3(1):1–20, 2006.

[17] R. van der Hofstad. Random graphs and complex networks, 2009.

[18] Y. Volkovich and N. Litvak. Asymptotic analysis for personalized web search. Adv. Appl. Prob., 42(2):577–604, 2010.

[19] Y. Volkovich, N. Litvak, and D. Donato. Determining factors behind the pagerank log-log plot. In Proceedings of the 5th International Conference on Algorithms and Models for the Web-graph, pages 108–123, 2007.

[20] L. Waltman and N.J. van Eck. The relation between eigenfactor, audience factor, and influence weight. J. Am. Soc. Inf. Sci., 61(7):1476–1486, 2010.